

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
る事項と同一であることを証明する。
This is to certify that the annexed is a true copy of the following application as filed
in this Office.

願 年 月 日 2 0 0 3 年 3 月 3 1 日
Date of Application:

願 番 号 特 願 2 0 0 3 - 0 9 6 1 2 9
Application Number:
[J P 2 0 0 3 - 0 9 6 1 2 9]
[T. 10/C]:

願 人 株 式 会 社 リ コ ー
Applicant(s):

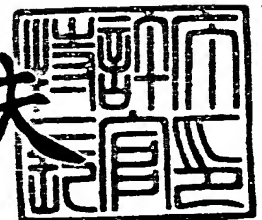
CERTIFIED COPY OF
PRIORITY DOCUMENT

BEST AVAILABLE COPY

2 0 0 4 年 4 月 1 2 日

特 許 庁 長 官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 0302634

【提出日】 平成15年 3月31日

【あて先】 特許庁長官 殿

【国際特許分類】 H04L 9/14

【発明の名称】 デジタル証明書管理システム、デジタル証明書管理装置、デジタル証明書管理方法、更新手順決定方法およびプログラム

【請求項の数】 28

【発明者】

【住所又は居所】 東京都大田区中馬込1丁目3番6号 株式会社リコー内

【氏名】 榎田 寛朗

【特許出願人】

【識別番号】 000006747

【住所又は居所】 東京都大田区中馬込1丁目3番6号

【氏名又は名称】 株式会社リコー

【代表者】 桜井 正光

【代理人】

【識別番号】 100080931

【住所又は居所】 東京都豊島区東池袋1丁目20番2号 池袋ホワイトハウスビル818号

【弁理士】

【氏名又は名称】 大澤 敬

【手数料の表示】

【予納台帳番号】 014498

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1



【物件名】 要約書 1

【包括委任状番号】 9809113

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 デジタル証明書管理システム、デジタル証明書管理装置、デジタル証明書管理方法、更新手順決定方法およびプログラム

【特許請求の範囲】

【請求項 1】 1 又は複数のクライアントと 1 又は複数のサーバとによって構成され、該各クライアントと各サーバとの間でデジタル証明書を用いて相互認証を行うようにしたクライアント・サーバシステムに、前記各クライアント及び前記各サーバと通信可能なデジタル証明書管理装置を接続したデジタル証明書管理システムであって、

前記デジタル証明書管理装置に、

前記各クライアント及び前記各サーバが前記相互認証に使用する前記デジタル証明書の正当性を確認するための証明鍵を更新する証明鍵更新手段と、

前記クライアント・サーバシステムを構成する各ノードについて、該ノードの通信相手及び該通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶する構成記憶手段と、

該構成記憶手段に記憶している情報をもとに、前記証明鍵更新手段による証明鍵の更新手順を制御する更新順制御手段とを設け、

前記証明鍵更新手段に、

更新用の新証明鍵を取得する手段と、

該新証明鍵を用いて正当性を確認可能な、前記相互認証に使用するための新デジタル証明書を取得する手段と、

前記各クライアントのための新デジタル証明書である新クライアント証明書と、前記新証明鍵とをそれぞれ対応する前記各クライアントに送信してこれを記憶するよう要求する第 1 の更新要求手段と、

前記各サーバのための新デジタル証明書である新サーバ証明書と、前記新証明鍵とをそれぞれ対応する前記各サーバに送信してこれを記憶するよう要求する第 2 の更新要求手段とを設け、

前記更新順制御手段が、前記第 2 の更新要求手段がそれぞれの前記サーバに前記新サーバ証明書を送信してこれを記憶するよう要求する動作を、該サーバの通

信相手となる全てのクライアントから前記新証明鍵を記憶した旨の応答があった後に行うように前記更新手順を制御する手段であることを特徴とするデジタル証明書管理システム。

【請求項 2】 請求項 1 記載のデジタル証明書管理システムであって、

前記デジタル証明書管理装置の前記更新順制御手段が、前記第 1 の更新要求手段がそれぞれの前記クライアントに前記新クライアント証明書を送信してこれを記憶するよう要求する動作を、該クライアントの通信相手となる全てのサーバから前記新証明鍵を記憶した旨の応答があった後に行うように前記更新手順を制御する手段であることを特徴とするデジタル証明書管理システム。

【請求項 3】 請求項 1 記載のデジタル証明書管理システムであって、

前記デジタル証明書管理装置の前記更新順制御手段が、前記第 1 の更新要求手段が前記新クライアント証明書と前記新証明鍵とを同時に前記各クライアントに送信してこれらを記憶するよう要求し、前記第 2 の更新要求手段が、それぞれの前記サーバに対して、該サーバの通信相手となる全てのクライアントから前記新証明鍵を記憶した旨の応答があった後で、前記新サーバ証明書と前記新証明鍵とを同時に送信し、これらを記憶するよう要求するよう前記更新手順を制御する手段であることを特徴とするデジタル証明書管理システム。

【請求項 4】 前記各サーバに、前記デジタル証明書管理装置と少なくとも一つの前記クライアントとの間の通信を仲介する手段を設け、

前記デジタル証明書管理装置と前記各クライアントとはいずれかの前記サーバを介して通信を行うことを特徴とする請求項 1 乃至 3 のいずれか一項記載のデジタル証明書管理システム。

【請求項 5】 前記各クライアントに、前記デジタル証明書管理装置と少なくとも一つの前記サーバとの間の通信を仲介する手段を設け、

前記デジタル証明書管理装置と前記各サーバとはいずれかの前記クライアントを介して通信を行うことを特徴とする請求項 1 乃至 3 のいずれか一項記載のデジタル証明書管理システム。

【請求項 6】 前記各クライアントに、前記デジタル証明書管理装置と該クライアントとの間の通信を仲介するサーバに対して定期的に通信を要求する手段

を設け、

前記サーバから前記クライアントへ送信すべき情報は、該通信の要求に対する応答として送信するようにしたことを特徴とする請求項 4 記載のデジタル証明書管理システム。

【請求項 7】 請求項 1 乃至 6 のいずれか一項記載のデジタル証明書管理システムであって、

前記デジタル証明書管理装置の前記証明鍵更新手段に、

従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む証明鍵証明書を取得する手段を設け、

前記第 1 の更新要求手段は、前記新証明鍵を前記証明鍵証明書の形式で前記各クライアントに送信してここに含まれる証明鍵を記憶するよう要求する手段であり、

前記第 2 の更新要求手段は、前記新証明鍵を前記証明鍵証明書の形式で前記各サーバに送信してここに含まれる証明鍵を記憶するよう要求する手段であり、

前記各クライアント及び前記各サーバにそれぞれ、

前記デジタル証明書管理装置から前記証明鍵証明書に含まれる証明鍵の記憶を要求された場合に、受信した証明鍵証明書の正当性を従前の証明鍵を用いて確認し、そこに含まれる証明鍵が適当なものであると判断した場合に該証明鍵を記憶する手段を設けたことを特徴とするデジタル証明書管理システム。

【請求項 8】 請求項 1 乃至 6 のいずれか一項記載のデジタル証明書管理システムであって、

前記デジタル証明書管理装置の前記証明鍵更新手段に、

従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第 1 の証明鍵証明書を取得する手段と、

前記新証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第 2 の証明鍵証明書を取得する手段とを設け、

前記第 1 の更新要求手段は、前記新証明鍵を前記第 1 及び第 2 の証明鍵証明書の形式でそれぞれ前記各クライアントに送信してこれを記憶するよう要求する手段であり、

前記第 2 の更新要求手段は、前記新証明鍵を前記第 1 及び第 2 の証明鍵証明書の形式でそれぞれ前記各サーバに送信してこれを記憶するよう要求する手段であり、

前記各クライアント及び前記各サーバにそれぞれ、

前記デジタル証明書管理装置から前記第 1 の証明鍵証明書を記憶するよう要求された場合に、該証明書の正当性を従前の証明鍵を用いて確認し、これが適当なものであると判断した場合に該証明書を記憶する手段と、

前記デジタル証明書管理装置から前記第 2 の証明鍵証明書を記憶するよう要求された場合に、該証明書の正当性を前記第 1 の証明鍵証明書に含まれる前記新証明鍵を用いて確認し、前記第 2 の証明鍵証明書が適当なものであると判断した場合に、該証明書を記憶すると共に従前の証明鍵証明書及び前記第 1 の証明鍵証明書を削除する手段とを設け、

前記デジタル証明書管理装置の前記更新順制御手段が、前記第 1 の更新要求手段が前記第 2 の証明鍵証明書をそれぞれの前記クライアントに送信してこれを記憶するよう要求する動作を、少なくとも該クライアントの通信相手となる全てのサーバから前記新サーバ証明書を記憶した旨の応答があった後に行うよう制御し、前記第 2 の更新要求手段が前記第 2 の証明鍵証明書をそれぞれの前記サーバに送信してこれを記憶するよう要求する動作を、少なくとも該サーバの通信相手となる全てのクライアントから前記新クライアント証明書を記憶した旨の応答があった後に行うように前記更新手順を制御する手段であることを特徴とするデジタル証明書管理システム。

【請求項 9】 請求項 1 乃至 8 のいずれか一項記載のデジタル証明書管理システムであって、

前記クライアントと前記サーバが行う前記相互認証は、SSL又はTLSのプロトコルに従った相互認証であり、

前記クライアント証明書及び前記サーバ証明書はそれぞれ前記各クライアント及び前記各サーバの公開鍵証明書であることを特徴とするデジタル証明書管理システム。

【請求項 10】 クライアント・サーバシステムを構成する 1 又は複数のク

クライアント及び1又は複数のサーバと通信可能なデジタル証明書管理装置であって、

前記各クライアントと前記各サーバとの間で相互認証に使用するデジタル証明書の正当性を確認するための証明鍵を更新する証明鍵更新手段と、

前記クライアント・サーバシステムを構成する各ノードについて、該ノードの通信相手及び該通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶する構成記憶手段と、

該構成記憶手段に記憶している情報をもとに、前記証明鍵更新手段による証明鍵の更新手順を制御する更新順制御手段とを設け、

前記証明鍵更新手段に、

更新用の新証明鍵を取得する手段と、

該新証明鍵を用いて正当性を確認可能な、前記相互認証に使用するための新デジタル証明書を取得する手段と、

前記クライアントのための新デジタル証明書である新クライアント証明書と、前記新証明鍵とをそれぞれ前記クライアントに送信してこれを記憶するよう要求する第1の更新要求手段と、

前記サーバのための新デジタル証明書である新サーバ証明書と、前記新証明鍵とをそれぞれ前記サーバに送信してこれを記憶するよう要求する第2の更新要求手段とを設け、

前記更新順制御手段が、前記第2の更新要求手段がそれぞれの前記サーバに対して前記新サーバ証明書を送信してこれを記憶するよう要求する動作を、該サーバの通信相手となる全てのクライアントからの前記新証明鍵を記憶した旨の応答があった後に行うように前記更新手順を制御する手段であることを特徴とするデジタル証明書管理装置。

【請求項11】 請求項10記載のデジタル証明書管理装置であって、

前記更新順制御手段が、前記第1の更新要求手段がそれぞれの前記クライアントに前記新クライアント証明書を送信してこれを記憶するよう要求する動作を、該クライアントの通信相手となる全てのサーバからの前記新証明鍵を記憶した旨の応答があった後に行うように前記更新手順を制御する手段であることを特徴と

するデジタル証明書管理装置。

【請求項 12】 請求項 10 記載のデジタル証明書管理装置であって、

前記更新順制御手段が、前記第 1 の更新要求手段が前記新クライアント証明書と前記新証明鍵とを同時に前記各クライアントに送信してこれらを記憶するよう要求し、前記第 2 の更新要求手段が、それぞれの前記サーバに対して、該サーバの通信相手となる全てのクライアントからの前記新証明鍵を記憶した旨の応答があった後で、前記新サーバ証明書と前記新証明鍵とを同時に送信し、これらを記憶するよう要求するように前記更新手順を制御する手段であることを特徴とするデジタル証明書管理装置。

【請求項 13】 請求項 10 乃至 12 のいずれか一項記載のデジタル証明書管理装置であって、

前記証明鍵更新手段に、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む証明鍵証明書を取得する手段を設け、

前記第 1 の更新要求手段は、前記新証明鍵を前記証明鍵証明書の形式で前記各クライアントに送信してここに含まれる証明鍵を記憶するよう要求する手段であり、

前記第 2 の更新要求手段は、前記新証明鍵を前記証明鍵証明書の形式で前記各サーバに送信してここに含まれる証明鍵を記憶するよう要求する手段であることを特徴とするデジタル証明書管理装置。

【請求項 14】 請求項 10 乃至 12 のいずれか一項記載のデジタル証明書管理装置であって、

前記証明鍵更新手段に、

従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第 1 の証明鍵証明書を取得する手段と、

前記新証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第 2 の証明鍵証明書を取得する手段とを設け、

前記第 1 の更新要求手段が、前記新証明鍵を前記第 1 及び第 2 の証明鍵証明書の形式でそれぞれ前記各クライアントに送信してこれを記憶するよう要求する手段であって、前記各クライアントに、前記第 2 の証明鍵証明書を記憶する場合に

従前の証明鍵証明書及び前記第 1 の証明鍵証明書を削除させる手段を有し、

前記第 2 の更新要求手段が、前記新証明鍵を前記第 1 及び第 2 の証明鍵証明書の形式でそれぞれ前記サーバに送信してこれを記憶するよう要求する手段であって、前記サーバに、前記第 2 の証明鍵証明書を記憶する場合には従前の証明鍵証明書及び前記第 1 の証明鍵証明書を削除させる手段を有し、

前記更新順制御手段が、前記第 1 の更新要求手段が前記第 2 の証明鍵証明書をそれぞれの前記クライアントに送信してこれを記憶するよう要求する動作を、少なくとも該クライアントの通信相手となる全てのサーバから前記新サーバ証明書を記憶した旨の応答があった後に行い、前記第 2 の更新要求手段が前記第 2 の証明鍵証明書をそれぞれの前記サーバに送信してこれを記憶するよう要求する動作を、少なくとも該サーバの通信相手となる全てのクライアントから前記新クライアント証明書を記憶した旨の応答があった後に行うように前記更新手順を制御する手段であることを特徴とするデジタル証明書管理装置。

【請求項 1 5】 請求項 1 0 乃至 1 4 のいずれか一項記載のデジタル証明書管理装置であって、

前記相互認証は、S S L 又は T L S のプロトコルに従った相互認証であり、

前記クライアント証明書及び前記サーバ証明書はそれぞれ前記各クライアント及び前記各サーバの公開鍵証明書であることを特徴とするデジタル証明書管理装置。

【請求項 1 6】 クライアント・サーバシステムを構成する 1 又は複数のクライアントと 1 又は複数のサーバとの間で相互認証に使用するデジタル証明書を、前記各クライアント及び前記各サーバと通信可能なデジタル証明書管理装置によって管理するデジタル証明書管理方法であって、

前記デジタル証明書管理装置が、

前記クライアント・サーバシステムを構成する各ノードについて、該ノードの通信相手及び該通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶しておき、該情報をもとに定める更新手順に従って前記各クライアント及び前記各サーバが前記相互認証に使用する前記デジタル証明書の正当性を確認するための証明鍵を更新し、

該証明鍵の更新を、

更新用の新証明鍵を取得する手順と、

該新証明鍵を用いて正当性を確認可能な、前記相互認証に使用するための新デジタル証明書を取得する手順と、

前記各クライアントのための新デジタル証明書である新クライアント証明書と、前記新証明鍵とをそれぞれ対応する前記各クライアントに送信してこれを記憶させる手順と、

前記各サーバのための新デジタル証明書である新サーバ証明書と、前記新証明鍵とをそれぞれ対応する前記各サーバに送信してこれを記憶させる手順とを実行することによって行い、

前記更新手順を、それぞれの前記サーバに前記新サーバ証明書を送信してこれを記憶させる手順を該サーバの通信相手となる全てのクライアントから前記新証明鍵を記憶した旨の応答があった後に行うよう定めることを特徴とするデジタル証明書管理方法。

【請求項 1 7】 請求項 1 6 記載のデジタル証明書管理方法であって、

前記更新手順を、それぞれの前記クライアントに前記新クライアント証明書を送信してこれを記憶させる手順を該クライアントの通信相手となる全てのサーバから前記新証明鍵を記憶した旨の応答があった後に行うよう定めることを特徴とするデジタル証明書管理方法。

【請求項 1 8】 請求項 1 6 記載のデジタル証明書管理方法であって、

前記更新手順を、前記新クライアント証明書と前記新証明鍵とを同時に前記各クライアントに送信してこれらを記憶させるように定め、さらに、それぞれの前記サーバに対して、該サーバの通信相手となる全てのクライアントから前記新証明鍵を記憶した旨の応答があった後で、前記新サーバ証明書と前記新証明鍵とを同時に送信し、これらを記憶させるように定めることを特徴とするデジタル証明書管理方法。

【請求項 1 9】 請求項 1 6 乃至 1 8 のいずれか一項記載のデジタル証明書管理方法であって、

前記証明鍵の更新の際に、

従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む証明鍵証明書を取得する手順をさらに実行し、

前記新証明鍵を前記各サーバあるいは前記各クライアントに送信してこれを記憶させる手順において、該新証明鍵を前記証明鍵証明書の形式で送信してここに含まれる証明鍵を記憶させるようにし、

前記各クライアント又は前記各サーバに前記証明鍵証明書に含まれる証明鍵を記憶させる場合に、該証明鍵証明書の正当性を、記憶している従前の証明鍵を用いて確認させ、そこに含まれる証明鍵が適当なものであると判断した場合に該証明鍵を記憶させることを特徴とするデジタル証明書管理方法。

【請求項 2 0】 請求項 1 6 乃至 1 8 のいずれか一項記載のデジタル証明書管理方法であって、

前記証明鍵の更新の際に、

従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第 1 の証明鍵証明書を取得する手順と、

前記新証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第 2 の証明鍵証明書を取得する手順とをさらに実行し、

前記新証明鍵を前記各サーバあるいは前記各クライアントに送信してこれを記憶させる手順において、該新証明鍵を前記第 1 及び第 2 の証明鍵証明書の形式でそれぞれ送信してこれを記憶させるようにし、

前記更新手順を、前記第 2 の証明鍵証明書をそれぞれの前記クライアントに送信してこれを記憶するよう要求する手順を少なくとも該クライアントの通信相手となる全てのサーバから前記新サーバ証明書を記憶した旨の応答があった後に行うよう定め、さらに、前記第 2 の証明鍵証明書をそれぞれの前記サーバに送信してこれを記憶するよう要求する動作を少なくとも該サーバの通信相手となる全てのクライアントから前記新クライアント証明書を記憶した旨の応答があった後に行うよう定め、

前記各クライアント又は前記各サーバに前記第 1 の証明鍵証明書を記憶させる際に、該証明書の正当性を従前の証明鍵を用いて確認させ、これが適当なものであると判断した場合に該証明書を記憶させ、

前記各クライアント又は前記各サーバに前記第 2 の証明鍵証明書を記憶させる際に、該証明書の正当性を前記第 1 の証明鍵証明書に含まれる前記新証明鍵を用いて確認させ、前記第 2 の証明鍵証明書が適当なものであると判断した場合に、該証明書を記憶させると共に従前の証明鍵証明書及び前記第 1 の証明鍵証明書を削除させることを特徴とするデジタル証明書管理方法。

【請求項 2 1】 請求項 1 6 乃至 2 0 のいずれか一項記載のデジタル証明書管理方法であって、

前記クライアントと前記サーバとの間の前記相互認証は、S S L 又は T L S のプロトコルに従った相互認証であり、

前記クライアント証明書及び前記サーバ証明書はそれぞれ前記各クライアント及び前記各サーバの公開鍵証明書であることを特徴とするデジタル証明書管理方法。

【請求項 2 2】 クライアント・サーバシステムを構成する 1 又は複数のクライアントと 1 又は複数のサーバとに記憶させ、これらの間で相互認証に使用するデジタル証明書の正当性を確認するための証明鍵を、前記各クライアント及び前記各サーバと通信可能なデジタル証明書管理装置によって更新する際の更新手順を定める更新手順決定方法であって、

前記デジタル証明書管理装置が、

前記クライアント・サーバシステムを構成する各ノードについて、該ノードの通信相手及び該通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶しておき、

該情報をもとに、前記更新手順を、

それぞれの前記サーバに、該サーバが前記相互認証に使用するための、更新用の新証明鍵を用いて正当性を確認可能な新デジタル証明書である前記新サーバ証明書を送信してこれを記憶させる手順を、該サーバの通信相手となる全てのクライアントから前記新証明鍵を記憶した旨の応答があった後に行うよう定めることを特徴とする更新手順決定方法。

【請求項 2 3】 クライアント・サーバシステムを構成する 1 又は複数のクライアント及び 1 又は複数のサーバと通信可能なデジタル証明書管理装置を制御

するコンピュータを、

前記各クライアントと前記各サーバとの間で相互認証に使用するデジタル証明書の正当性を確認するための証明鍵を更新する証明鍵更新手段と、

前記クライアント・サーバシステムを構成する各ノードについて、該ノードの通信相手及び該通信相手との間でクライアントとサーバのいずれとして機能するか情報を記憶する構成記憶手段と、

該構成記憶手段に記憶している情報をもとに、前記証明鍵更新手段による証明鍵の更新手順を制御する更新順制御手段として機能させるためのプログラムであって、

前記証明鍵更新手段は、

更新用の新証明鍵を取得する手段と、

該新証明鍵を用いて正当性を確認可能な、前記相互認証に使用するための新デジタル証明書を取得する手段と、

前記クライアントのための新デジタル証明書である新クライアント証明書と、前記新証明鍵とをそれぞれ前記クライアントに送信してこれを記憶するよう要求する第 1 の更新要求手段と、

前記サーバのための新デジタル証明書である新サーバ証明書と、前記新証明鍵とをそれぞれ前記サーバに送信してこれを記憶するよう要求する第 2 の更新要求手段との機能を有し、

前記更新順制御手段が、前記第 2 の更新要求手段がそれぞれの前記サーバに対して前記新サーバ証明書を送信してこれを記憶するよう要求する動作を、該サーバの通信相手となる全てのクライアントからの前記新証明鍵を記憶した旨の応答があった後に行うように前記更新手順を制御するようにしたことを特徴とするプログラム。

【請求項 2 4】 請求項 2 3 記載のプログラムであって、

前記更新順制御手段が、前記第 1 の更新要求手段がそれぞれの前記クライアントに前記新クライアント証明書を送信してこれを記憶するよう要求する動作を、該クライアントの通信相手となる全てのサーバからの前記新証明鍵を記憶した旨の応答があった後に行うように前記更新手順を制御するようにしたことを特徴と

するプログラム。

【請求項 25】 請求項 23 記載のプログラムであって、

前記更新順制御手段が、前記第 1 の更新要求手段が前記新クライアント証明書と前記新証明鍵とを同時に前記各クライアントに送信してこれらを記憶するよう要求し、前記第 2 の更新要求手段が、それぞれの前記サーバに対して、該サーバの通信相手となる全てのクライアントからの前記新証明鍵を記憶した旨の応答があった後で、前記新サーバ証明書と前記新証明鍵とを同時に前記サーバに送信し、これらを記憶するよう要求するように前記更新手順を制御するようにしたことを特徴とするプログラム。

【請求項 26】 請求項 23 乃至 25 のいずれか一項記載のプログラムであって、

前記コンピュータを、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む証明鍵証明書を取得する手段として機能させるためのプログラムをさらに含み、

前記第 1 の更新要求手段が、前記新証明鍵を前記証明鍵証明書の形式で前記各クライアントに送信してここに含まれる証明鍵を記憶するよう要求するようにし、

前記第 2 の更新要求手段が、前記新証明鍵を前記証明鍵証明書の形式で前記各サーバに送信してここに含まれる証明鍵を記憶するよう要求するようにしたことを特徴とするプログラム。

【請求項 27】 請求項 23 乃至 25 のいずれか一項記載のプログラムであって、

前記コンピュータを、

従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第 1 の証明鍵証明書を取得する手段と、

前記新証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第 2 の証明鍵証明書を取得する手段として機能させるためのプログラムをさらに含み、

前記第 1 の更新要求手段が、前記新証明鍵を前記第 1 及び第 2 の証明鍵証明書

の形式でそれぞれ前記各クライアントに送信してこれを記憶するよう要求し、前記各クライアントに、前記第2の証明鍵証明書を記憶する場合には従前の証明鍵証明書及び前記第1の証明鍵証明書を削除させる機能を有し、

前記第2の更新要求手段が、前記新証明鍵を前記第1及び第2の証明鍵証明書の形式でそれぞれ前記各サーバに送信してこれを記憶するよう要求し、前記各サーバに、前記第2の証明鍵証明書を記憶する場合には従前の証明鍵証明書及び前記第1の証明鍵証明書を削除させる機能を有し、

前記更新順制御手段が、前記第1の更新要求手段が前記第2の証明鍵証明書をそれぞれの前記クライアントに送信してこれを記憶するよう要求する動作を、少なくとも該クライアントの通信相手となる全てのサーバから前記新サーバ証明書を記憶した旨の応答があった後に行い、前記第2の更新要求手段が前記第2の証明鍵証明書をそれぞれの前記サーバに送信してこれを記憶するよう要求する動作を、少なくとも該サーバの通信相手となる全てのクライアントから前記新クライアント証明書を記憶した旨の応答があった後に行うように前記更新手順を制御するようにしたことを特徴とするプログラム。

【請求項28】 請求項23乃至27のいずれか一項記載のプログラムであって、

前記相互認証は、SSL又はTLSのプロトコルに従った相互認証であり、

前記クライアント証明書及び前記サーバ証明書はそれぞれ前記各クライアント及び前記各サーバの公開鍵証明書であることを特徴とするプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、デジタル証明書管理装置によってクライアント・サーバシステムを構成する1又は複数のクライアントと1又は複数のサーバの間の認証処理に用いるデジタル証明書を管理するデジタル証明書管理システム、このようなシステムを構成するデジタル証明書管理装置、このようにデジタル証明書を管理するデジタル証明書管理方法、このデジタル証明書の管理に際してそのデジタル証明書の正当性を確認するための証明鍵を更新する場合の更新手順決定方法、およびコ

ンピュータを上記のデジタル証明書管理装置として機能させるためのプログラムに関する。

【0002】

【従来の技術】

従来から、PC等のコンピュータを複数台ネットワークを介して通信可能に接続し、少なくとも1台をサーバ装置（サーバ）、別の少なくとも1台をクライアント装置（クライアント）としたクライアント・サーバシステムを構成することが行われている。

このようなクライアント・サーバシステムにおいては、クライアント装置からサーバ装置に要求を送信し、サーバ装置がその要求に従った処理を行ってクライアント装置に対して応答を返す。そして、このようなクライアント・サーバシステムは、クライアント装置から商品の注文要求を送信し、サーバ装置においてその注文を受け付けるといった、いわゆる電子商取引にも広く用いられるようになっている。また、種々の電子装置にクライアント装置あるいはサーバ装置の機能を持たせてネットワークを介して接続し、相互間の通信によって電子装置の遠隔管理を行うシステムも提案されている。

【0003】

このような場合においては、通信相手が適切か、あるいは送信される情報が改竄されていないかといった確認が重要である。また、特にインターネットにおいては、情報が通信相手に到達するまでに無関係なコンピュータを経由する場合が多いことから、機密情報を送信する場合、その内容を盗み見られないようにする必要もある。そして、このような要求に応える通信プロトコルとして、例えばSSL（Secure Socket Layer）と呼ばれるプロトコルが開発されており、広く用いられている。このプロトコルを用いて通信を行うことにより、公開鍵暗号方式と共通鍵暗号方式とを組み合わせ、通信相手の認証を行うと共に、情報の暗号化により改竄及び盗聴の防止を図ることができる。

【0004】

ここで、このSSLを用いて相互認証を行う場合の通信手順について、認証処理の部分に焦点を当てて説明する。図46は、クライアント装置とサーバ装置と

がSSLによる相互認証を行う際の各装置において実行する処理のフローチャートを、その処理に用いる情報と共に示す図である。

図46に示すように、SSLによる相互認証を行う際には、まずクライアント装置側にルート鍵証明書、クライアント私有鍵、クライアント公開鍵証明書（クライアント証明書）を記憶させておく必要がある。クライアント私有鍵は、認証局（CA：certificate authority）がクライアント装置に対して発行した私有鍵である。そして、クライアント公開鍵証明書は、その私有鍵と対応する公開鍵にCAがデジタル署名を付してデジタル証明書としたものである。また、ルート鍵証明書は、CAがデジタル署名に用いた証明用私有鍵であるルート私有鍵と対応する証明用公開鍵（以下「証明鍵」ともいう）であるルート鍵に、デジタル署名を付してデジタル証明書としたものである。

【0005】

図47にこれらの関係を示す。

図47（a）に示すように、クライアント公開鍵は、クライアント私有鍵を用いて暗号化された文書を復号化するための鍵本体と、その公開鍵の発行者（CA）、発行相手（クライアント装置）、有効期限等の情報を含む書誌情報とによって構成される。そして、CAは、鍵本体や書誌情報が改竄されていないことを示すため、クライアント公開鍵をハッシュ処理して得たハッシュ値を、ルート私有鍵を用いて暗号化し、デジタル署名としてクライアント公開鍵に付す。またこの際に、デジタル署名に用いるルート私有鍵の識別情報を署名鍵情報として公開鍵の書誌情報に加える。そして、このデジタル署名を付した公開鍵証明書が、クライアント公開鍵証明書である。

【0006】

このクライアント公開鍵証明書を認証処理に用いる場合には、ここに含まれるデジタル署名を、ルート私有鍵と対応する公開鍵であるルート鍵の鍵本体を用いて復号化する。この復号化が正常に行われれば、デジタル署名が確かにCAによって付されたことがわかる。また、クライアント公開鍵部分をハッシュ処理して得たハッシュ値と、復号して得たハッシュ値とが一致すれば、鍵自体も損傷や改竄を受けていないことがわかる。さらに、受信したデータをこのクライアント公

開鍵を用いて正常に復号化できれば、そのデータは、クライアント私有鍵の持ち主、つまりクライアント装置から送信されたものであることがわかる。あとは、書誌情報を参照して、CAの信頼性やクライアント装置の登録有無等によって認証の正否を決定すればよい。

【0 0 0 7】

ここで、認証を行うためには、ルート鍵を予め記憶しておく必要があるが、このルート鍵も、図47(b)に示すように、CAがデジタル署名を付したルート鍵証明書として記憶しておく。このルート鍵証明書は、自身に含まれる公開鍵でデジタル署名を復号化可能な、自己署名形式である。そして、ルート鍵を使用する際に、そのルート鍵証明書に含まれる鍵本体を用いてデジタル署名を復号化し、ルート鍵をハッシュ処理して得たハッシュ値と比較する。これが一致すれば、ルート鍵が破損等していないことを確認できるのである。

【0 0 0 8】

図46の説明に戻ると、サーバ装置側には、ルート鍵証明書、サーバ私有鍵、サーバ公開鍵証明書（サーバ証明書）を記憶させておく必要がある。サーバ私有鍵及びサーバ公開鍵証明書は、CAがサーバ装置に対して発行した私有鍵及び公開鍵証明書である。ここではクライアント装置とサーバ装置に対して同じCAが同じルート私有鍵を用いて証明書を発行しているものとし、この場合にはルート鍵証明書はクライアント装置とサーバ装置で共通となる。

【0 0 0 9】

フローチャートの説明に入る。なお、図46において、2本のフローチャート間の矢印は、データの転送を示し、送信側は矢印の根元のステップで転送処理を行い、受信側はその情報を受信すると矢印の先端のステップの処理を行うものとする。また、各ステップの処理が正常に完了しなかった場合には、その時点で認証失敗の応答を返して処理を中断するものとする。相手から認証失敗の応答を受けた場合、処理がタイムアウトした場合等も同様である。

【0 0 1 0】

クライアント・サーバシステムにおいて、接続を要求するのはクライアント装置側であるが、ユーザの指示等によってこの必要が生じた場合、クライアント装

置のCPUは、所要の制御プログラムを実行することにより、図46の左側に示すフローチャートの処理を開始する。そして、ステップS11でサーバ装置に対して接続要求を送信する。

一方サーバ装置のCPUは、この接続要求を受信すると、所要の制御プログラムを実行することにより、図46の右側に示すフローチャートの処理を開始する。そして、ステップS21で第1の乱数を生成し、これをサーバ私有鍵を用いて暗号化する。そして、ステップS22でその暗号化した第1の乱数とサーバ公開鍵証明書とをクライアント装置に送信する。このステップS22の処理において、サーバ装置のCPUが第1のサーバ側認証処理手段として機能する。

【0011】

クライアント装置側では、これを受信すると、ステップS12でルート鍵証明書を用いてサーバ公開鍵証明書の正当性を確認する。これには、上述のように損傷や改竄を受けていないことを確認するのみならず、書誌情報を参照してサーバ装置が適当な通信相手であることを確認する処理を含む。

そして確認ができると、ステップS13で、受信したサーバ公開鍵証明書に含まれるサーバ公開鍵を用いて第1の乱数を復号化する。ここで復号化が成功すれば、第1の乱数は確かにサーバ公開鍵証明書の発行対象であるサーバ装置から受信したものだ確認できる。そして、サーバ装置を正当な通信相手として認証する。このステップS12及びS13の処理において、クライアント装置のCPUが第2のクライアント側認証処理手段として機能する。

【0012】

その後、ステップS14でこれとは別に第2の乱数及び第3の乱数を生成する。そして、ステップS15で第2の乱数をクライアント私有鍵を用いて暗号化し、第3の乱数をサーバ公開鍵を用いて暗号化し、ステップS16でこれらをクライアント公開鍵証明書と共にサーバ装置に送信する。第3の乱数の暗号化は、サーバ装置以外の装置に乱数を知られないようにするために行うものである。このステップS16の処理において、クライアント装置のCPUが第1のクライアント側認証処理手段として機能する。

【0013】

サーバ装置側では、これを受信すると、ステップS23でルート鍵証明書を用いてクライアント公開鍵証明書の正当性を確認する。これにも、ステップS12の場合と同様、クライアント装置が適当な通信相手であることを確認する処理を含む。そして確認ができると、ステップS24で、受信したクライアント公開鍵証明書に含まれるクライアント公開鍵を用いて第2の乱数を復号化する。ここで復号化が成功すれば、第2の乱数は確かにクライアント公開鍵証明書の発行対象であるクライアント装置から受信したものだ確認できる。そして、サーバ装置を正当な通信相手として認証する。このステップS23及び24の処理において、サーバ装置のCPUが第2のサーバ側認証処理手段として機能する。

その後、ステップS25でサーバ私有鍵を用いて第3の乱数を復号化する。ここまでの処理で、サーバ側とクライアント側に共通の第1乃至第3の乱数が共有されたことになる。そして、少なくとも第3の乱数は、生成したクライアント装置と、サーバ私有鍵を持つサーバ装置以外の装置が知ることはない。ここまでの処理が成功すると、ステップS26でクライアント装置に対して認証成功の応答を返す。

【0014】

クライアント装置側では、これを受信すると、ステップS17で第1乃至第3の乱数から共通鍵を生成し、以後の通信の暗号化に用いるものとして認証処理を終了する。サーバ装置側でも、ステップS27で同様の処理を行って終了する。そして、以上の処理によって互いに通信を確立し、以後はステップS17又はS27で生成した共通鍵を用い、共通鍵暗号方式でデータを暗号化して通信を行う。

このような処理を行うことにより、クライアント装置とサーバ装置が互いに相手を認証した上で安全に共通鍵を交換することができ、通信を確かな相手と安全に行うことができる。

【0015】

ところで、公開鍵暗号方式においては、鍵長にもよるが、時間をかければ公開鍵から私有鍵を導くことができる。そして、私有鍵がわかってしまえば、第3者がその私有鍵の持ち主になりすますことが可能になるので、認証の確実性や通信

の安全性が保たれない。そこで、上述のように鍵に有効期限を設け、所定期間毎に鍵のセットを更新するというセキュリティポリシーを採用するユーザが増えている。このため、例えば上記のような相互認証を利用した遠隔管理システム等を提供する場合には、顧客に対し、鍵の更新が可能なシステムであるという保証を行う必要が生じている。これは、ルート鍵とルート私有鍵についても同様である。なお、鍵の更新事由としては、所定の有効期限の到来の他にも、私有鍵の第3者への漏洩が判明した場合等が考えられる。

このような鍵の更新に関する技術としては、例えば特許文献1に記載のものが挙げられる。

【0016】

【特許文献1】

特開平11-122238号公報

【0017】

【発明が解決しようとする課題】

しかしながら、特許文献1には、各装置に対して発行した鍵の更新に関する記載はあるが、ルート鍵の更新についての記載はない。

公開鍵暗号方式の場合、各装置に発行した鍵のペアを更新する場合には、その装置には新たな私有鍵に対応した新たな公開鍵証明書が記憶されることになり、通信相手にこれを渡せば、図46に示した認証処理を支障なく行うことができる。

しかし、ルート鍵を更新する場合、新たなルート鍵では従前のデジタル証明書に付されたデジタル署名を復号化することができないため、新たなルート鍵と対応する新たなルート私有鍵を用いて各装置の公開鍵証明書を作成し直し、これを配布しなければ、図46に示した認証処理の実行に支障を来してしまう（ただし、各装置の私有鍵は必ずしも更新する必要はない）。

【0018】

そして、認証処理に支障を来さずにルート鍵を更新する方式が知られていなかったため、更新の必要な装置にルート鍵をネットワークを介して安全に送信することができなかった。そこで、ルート鍵証明書や新たな公開鍵証明書を別の安全

な経路で各装置に届ける必要があったのである。

この経路としては、例えば書留郵便が考えられ、証明書のデータを記録したメモリカードやフレキシブルディスク等の記録媒体を装置の管理者に書留郵便で送付し、管理者が装置の鍵を更新するという方式が考えられる。しかし、この方式では、クライアントやサーバの各装置について十分な知識を持った管理者がいる場合にしか適用できないし、CA側は記録媒体を送付した後の処理については装置の管理者を信用するしかなかった。従って、管理者が更新処理を怠ったり誤ったりした場合には、認証処理が行えなくなってしまうという問題があった。

【0019】

一方管理者側も、受け取った証明書が正しいものであるか否かは、封筒やデータに記載された送り主の名称等を信用して判断するしかなく、CAの名を騙る別人から受け取った二重の証明書を装置に記憶させてしまうといった危険は常につきまとうことになる。

また、CAやクライアント・サーバシステムによるサービスの提供者が、各装置の配置先にサービスマンを派遣して鍵の更新を行うことも考えられるが、広い地域でこのような方式を採るには多数のサービス拠点が必要になり、コストが嵩むことになる。また、サービスマンの教育や不正防止、更新作業用の管理者IDの管理も問題となる。例えば、認証情報を手入力する単純な方式を採ろうとすると、退職したサービスマンについての更新権限を抹消するためには、各装置に記憶させている認証情報を変更する必要があるが、顧客先に設置された多数の装置にこのような変更を行うことは困難である。

【0020】

結局のところ、ネットワークを介さずに証明書の安全な配布経路を確保するためには、人間を信用する他なく、そこには欺瞞が入りこむ余地が出てしまう。そして、この余地を小さくするよう管理することはできるが、そのためには膨大なコストが生じてしまい、欺瞞の危険を考慮しなくて済むレベルの経路を証明書の配布のために構築することは、現実的ではなかった。

従来の技術の項で述べたとおり、SSLを用いた相互認証は理論的には可能なのであるが、以上のような問題があり、ルート鍵の安全な更新が実質的に不可能

であったので、実際には使用されていないのが現状である。またこのような問題は、公開鍵暗号とデジタル証明書を用いて認証を行う他のプロトコルを用いた場合にも、同様に発生するものと考えられる。

【0021】

この発明は、このような問題を解決し、クライアント・サーバシステムにおける認証処理でデジタル証明書の正当性を確認するために用いる証明鍵を、自動的に更新できるようにすることを目的とする。そして、このことにより、公開鍵暗号を利用したデジタル証明書を用いるSSL等の方式による相互認証を、クライアント・サーバシステムにおいて低コストで実現可能とすることを目的とする。

【0022】

【課題を解決するための手段】

上記の目的を達成するため、この発明のデジタル証明書管理装置は、1又は複数のクライアントと1又は複数のサーバとによって構成され、その各クライアントと各サーバとの間でデジタル証明書を用いて相互認証を行うようにしたクライアント・サーバシステムに、上記各クライアント及び上記各サーバと通信可能なデジタル証明書管理装置を接続したデジタル証明書管理システムにおいて、上記デジタル証明書管理装置に、上記各クライアント及び上記各サーバが上記相互認証に使用する上記デジタル証明書の正当性を確認するための証明鍵を更新する証明鍵更新手段と、上記クライアント・サーバシステムを構成する各ノードについて、そのノードの通信相手及びその通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶する構成記憶手段と、その構成記憶手段に記憶している情報をもとに、上記証明鍵更新手段による証明鍵の更新手順を制御する更新順制御手段とを設け、上記証明鍵更新手段に、更新用の新証明鍵を取得する手段と、その新証明鍵を用いて正当性を確認可能な、上記相互認証に使用するための新デジタル証明書を取得する手段と、上記各クライアントのための新デジタル証明書である新クライアント証明書と、上記新証明鍵とをそれぞれ対応する上記各クライアントに送信してこれを記憶するよう要求する第1の更新要求手段と、上記各サーバのための新デジタル証明書である新サーバ証明書と、上記新証明鍵とをそれぞれ対応する上記各サーバに送信してこれを記憶するよう要求する

第 2 の更新要求手段とを設け、上記更新順制御手段を、上記第 2 の更新要求手段がそれぞれの上記サーバに上記新サーバ証明書を送信してこれを記憶するよう要求する動作を、そのサーバの通信相手となる全てのクライアントから上記新証明鍵を記憶した旨の応答があった後に行うように上記更新手順を制御する手段としたものである。

【 0 0 2 3 】

このようなデジタル証明書管理システムにおいて、上記デジタル証明書管理装置の上記更新順制御手段を、上記第 1 の更新要求手段がそれぞれの上記クライアントに上記新クライアント証明書を送信してこれを記憶するよう要求する動作を、そのクライアントの通信相手となる全てのサーバから上記新証明鍵を記憶した旨の応答があった後に行うように上記更新手順を制御する手段とするとよい。

あるいは、上記デジタル証明書管理装置の上記更新順制御手段を、上記第 1 の更新要求手段が上記新クライアント証明書と上記新証明鍵とを同時に上記各クライアントに送信してこれらを記憶するよう要求し、上記第 2 の更新要求手段が、それぞれの上記サーバに対して、そのサーバの通信相手となる全てのクライアントから上記新証明鍵を記憶した旨の応答があった後で、上記新サーバ証明書と上記新証明鍵とを同時に送信し、これらを記憶するよう要求するように上記更新手順を制御する手段とするとよい。

【 0 0 2 4 】

これらのデジタル証明書管理システムにおいて、上記各サーバに、上記デジタル証明書管理装置と少なくとも一つの上記クライアントとの間の通信を仲介する手段を設け、上記デジタル証明書管理装置と上記各クライアントとはいずれかの上記サーバを介して通信を行うようにするとよい。

さらに、上記各クライアントに、上記デジタル証明書管理装置とそのクライアントとの間の通信を仲介するサーバに対して定期的に通信を要求する手段を設け、上記サーバから上記クライアントへ送信すべき情報は、その通信の要求に対する応答として送信するようにするとよい。

あるいは、上記のこれらのデジタル証明書管理システムにおいて、上記各クライアントに、上記デジタル証明書管理装置と少なくとも一つの上記サーバとの間

の通信を仲介する手段を設け、上記デジタル証明書管理装置と上記各サーバとはいずれかの上記クライアントを介して通信を行うようにするとよい。

【 0 0 2 5 】

さらに、これらのデジタル証明書管理システムにおいて、上記デジタル証明書管理装置の上記証明鍵更新手段に、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む証明鍵証明書を取得する手段を設け、上記第 1 の更新要求手段を、上記新証明鍵を上記証明鍵証明書の形式で上記各クライアントに送信してここに含まれる証明鍵を記憶するよう要求する手段とし、上記第 2 の更新要求手段を、上記新証明鍵を上記証明鍵証明書の形式で上記各サーバに送信してここに含まれる証明鍵を記憶するよう要求する手段とし、上記各クライアント及び上記各サーバにそれぞれ、上記デジタル証明書管理装置から上記証明鍵証明書に含まれる証明鍵の記憶を要求された場合に、受信した証明鍵証明書の正当性を従前の証明鍵を用いて確認し、そこに含まれる証明鍵が適当なものであると判断した場合にその証明鍵を記憶する手段を設けるとよい。

【 0 0 2 6 】

あるいは、上記デジタル証明書管理装置の上記証明鍵更新手段に、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む第 1 の証明鍵証明書を取得する手段と、上記新証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む第 2 の証明鍵証明書を取得する手段とを設け、上記第 1 の更新要求手段を、上記新証明鍵を上記第 1 及び第 2 の証明鍵証明書の形式でそれぞれ上記各クライアントに送信してこれを記憶するよう要求する手段とし、上記第 2 の更新要求手段を、上記新証明鍵を上記第 1 及び第 2 の証明鍵証明書の形式でそれぞれ上記各サーバに送信してこれを記憶するよう要求する手段とし、上記各クライアント及び上記各サーバにそれぞれ、上記デジタル証明書管理装置から上記第 1 の証明鍵証明書を記憶するよう要求された場合に、その証明書の正当性を従前の証明鍵を用いて確認し、これが適当なものであると判断した場合にその証明書を記憶する手段と、上記デジタル証明書管理装置から上記第 2 の証明鍵証明書を記憶するよう要求された場合に、その証明書の正当性を上記第 1 の証明鍵証明書に含まれる上記新証明鍵を用いて確認し、上記第 2 の

証明鍵証明書が適当なものであると判断した場合に、その証明書を記憶すると共に従前の証明鍵証明書及び上記第1の証明鍵証明書を削除する手段とを設け、上記デジタル証明書管理装置の上記更新順制御手段を、上記第1の更新要求手段が上記第2の証明鍵証明書をそれぞれの上記クライアントに送信してこれを記憶するよう要求する動作を、少なくともそのクライアントの通信相手となる全てのサーバから上記新サーバ証明書を記憶した旨の応答があった後に行うよう制御し、上記第2の更新要求手段が上記第2の証明鍵証明書をそれぞれの上記サーバに送信してこれを記憶するよう要求する動作を、少なくともそのサーバの通信相手となる全てのクライアントから上記新クライアント証明書を記憶した旨の応答があった後に行うように上記更新手順を制御する手段とするとよい。

【0027】

さらに、上記クライアントと上記サーバが行う上記相互認証を、SSL又はTLSのプロトコルに従った相互認証とし、上記クライアント証明書及び上記サーバ証明書をそれぞれ上記各クライアント及び上記各サーバの公開鍵証明書とする
とよい。

【0028】

また、この発明のデジタル証明書管理装置は、クライアント・サーバシステムを構成する1又は複数のクライアント及び1又は複数のサーバと通信可能なデジタル証明書管理装置において、上記各クライアントと上記各サーバとの間で相互認証に使用するデジタル証明書の正当性を確認するための証明鍵を更新する証明鍵更新手段と、上記クライアント・サーバシステムを構成する各ノードについて、そのノードの通信相手及びその通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶する構成記憶手段と、その構成記憶手段に記憶している情報をもとに、上記証明鍵更新手段による証明鍵の更新手順を制御する更新順制御手段とを設け、上記証明鍵更新手段に、更新用の新証明鍵を取得する手段と、その新証明鍵を用いて正当性を確認可能な、上記相互認証に使用するための新デジタル証明書を取得する手段と、上記クライアントのための新デジタル証明書である新クライアント証明書と、上記新証明鍵とをそれぞれ上記クライアントに送信してこれを記憶するよう要求する第1の更新要求手段と、上記サーバ

のための新デジタル証明書である新サーバ証明書と、上記新証明鍵とをそれぞれ上記サーバに送信してこれを記憶するよう要求する第 2 の更新要求手段とを設け、上記更新順制御手段を、上記第 2 の更新要求手段がそれぞれの上記サーバに対して上記新サーバ証明書を送信してこれを記憶するよう要求する動作を、そのサーバの通信相手となる全てのクライアントからの上記新証明鍵を記憶した旨の応答があった後に行うように上記更新手順を制御する手段としたものである。

【 0 0 2 9 】

このようなデジタル証明書管理装置において、上記更新順制御手段を、上記第 1 の更新要求手段がそれぞれの上記クライアントに上記新クライアント証明書を送信してこれを記憶するよう要求する動作を、そのクライアントの通信相手となる全てのサーバからの上記新証明鍵を記憶した旨の応答があった後に行うように上記更新手順を制御する手段とするとよい。

さらに、上記更新順制御手段を、上記第 1 の更新要求手段が上記新クライアント証明書と上記新証明鍵とを同時に上記各クライアントに送信してこれらを記憶するよう要求し、上記第 2 の更新要求手段が、それぞれの上記サーバに対して、そのサーバの通信相手となる全てのクライアントからの上記新証明鍵を記憶した旨の応答があった後で、上記新サーバ証明書と上記新証明鍵とを同時に送信し、これらを記憶するよう要求するように上記更新手順を制御する手段とするとよい。

【 0 0 3 0 】

これらのデジタル証明書管理装置において、上記証明鍵更新手段に、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む証明鍵証明書を取得する手段を設け、上記第 1 の更新要求手段を、上記新証明鍵を上記証明鍵証明書の形式で上記各クライアントに送信してここに含まれる証明鍵を記憶するよう要求する手段とし、上記第 2 の更新要求手段を、上記新証明鍵を上記証明鍵証明書の形式で上記各サーバに送信してここに含まれる証明鍵を記憶するよう要求する手段とするとよい。

【 0 0 3 1 】

あるいは、上記証明鍵更新手段に、従前の証明鍵を用いて正当性を確認可能な

デジタル証明書であって上記新証明鍵を含む第 1 の証明鍵証明書を取得する手段と、上記新証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む第 2 の証明鍵証明書を取得する手段とを設け、上記第 1 の更新要求手段が、上記新証明鍵を上記第 1 及び第 2 の証明鍵証明書の形式でそれぞれ上記各クライアントに送信してこれを記憶するよう要求する手段であって、上記各クライアントに、上記第 2 の証明鍵証明書を記憶する場合に従前の証明鍵証明書及び上記第 1 の証明鍵証明書を削除させる手段を有するようにし、上記第 2 の更新要求手段を、上記新証明鍵を上記第 1 及び第 2 の証明鍵証明書の形式でそれぞれ上記サーバに送信してこれを記憶するよう要求する手段であって、上記サーバに、上記第 2 の証明鍵証明書を記憶する場合には従前の証明鍵証明書及び上記第 1 の証明鍵証明書を削除させる手段を有するようにし、上記更新順制御手段を、上記第 1 の更新要求手段が上記第 2 の証明鍵証明書をそれぞれの上記クライアントに送信してこれを記憶するよう要求する動作を、少なくともそのクライアントの通信相手となる全てのサーバから上記新サーバ証明書を記憶した旨の応答があった後に行い、上記第 2 の更新要求手段が上記第 2 の証明鍵証明書をそれぞれの上記サーバに送信してこれを記憶するよう要求する動作を、少なくともそのサーバの通信相手となる全てのクライアントから上記新クライアント証明書を記憶した旨の応答があった後に行うように上記更新手順を制御する手段とするとよい。

【0032】

これらのデジタル証明書管理装置において、上記相互認証を、SSL又はTLSのプロトコルに従った相互認証とし、上記クライアント証明書及び上記サーバ証明書をそれぞれ上記各クライアント及び上記各サーバの公開鍵証明書とするとよい。

【0033】

また、この発明のデジタル証明書管理方法は、クライアント・サーバシステムを構成する 1 又は複数のクライアントと 1 又は複数のサーバとの間で相互認証に使用するデジタル証明書を、上記各クライアント及び上記各サーバと通信可能なデジタル証明書管理装置によって管理するデジタル証明書管理方法において、上記デジタル証明書管理装置を、上記クライアント・サーバシステムを構成する各

ノードについて、そのノードの通信相手及びその通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶しておき、その情報をもとに定める更新手順に従って上記各クライアント及び上記各サーバが上記相互認証に使用する上記デジタル証明書の正当性を確認するための証明鍵を更新し、その証明鍵の更新を、更新用の新証明鍵を取得する手順と、その新証明鍵を用いて正当性を確認可能な、上記相互認証に使用するための新デジタル証明書を取得する手順と、上記各クライアントのための新デジタル証明書である新クライアント証明書と、上記新証明鍵とをそれぞれ対応する上記各クライアントに送信してこれを記憶させる手順と、上記各サーバのための新デジタル証明書である新サーバ証明書と、上記新証明鍵とをそれぞれ対応する上記各サーバに送信してこれを記憶させる手順とを実行することによって行い、上記更新手順を、それぞれの上記サーバに上記新サーバ証明書を送信してこれを記憶させる手順をそのサーバの通信相手となる全てのクライアントから上記新証明鍵を記憶した旨の応答があった後に行うよう定めるようにするとよい。

【0034】

このようなデジタル証明書管理方法において、上記更新手順を、それぞれの上記クライアントに上記新クライアント証明書を送信してこれを記憶させる手順をそのクライアントの通信相手となる全てのサーバから上記新証明鍵を記憶した旨の応答があった後に行うよう定めるようにするとよい。

あるいは、上記更新手順を、上記新クライアント証明書と上記新証明鍵とを同時に上記各クライアントに送信してこれらを記憶させるように定め、さらに、それぞれの上記サーバに対して、そのサーバの通信相手となる全てのクライアントから上記新証明鍵を記憶した旨の応答があった後で、上記新サーバ証明書と上記新証明鍵とを同時に送信し、これらを記憶させるように定めるようにするとよい。

【0035】

これらのデジタル証明書管理方法において、上記証明鍵の更新の際に、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む証明鍵証明書を取得する手順をさらに実行し、上記新証明鍵を上記各サーバある

いは上記各クライアントに送信してこれを記憶させる手順において、その新証明鍵を上記証明鍵証明書で送信してここに含まれる証明鍵を記憶させるようにし、上記各クライアント又は上記各サーバに上記証明鍵証明書に含まれる証明鍵を記憶させる場合に、その証明鍵証明書の正当性を、記憶している従前の証明鍵を用いて確認させ、そこに含まれる証明鍵が適当なものであると判断した場合にその証明鍵を記憶させるようにするとよい。

【0036】

あるいは、上記証明鍵の更新の際に、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む第1の証明鍵証明書を取得する手順と、上記新証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む第2の証明鍵証明書を取得する手順とをさらに実行し、上記新証明鍵を上記各サーバあるいは上記各クライアントに送信してこれを記憶させる手順において、その新証明鍵を上記第1及び第2の証明鍵証明書の形式でそれぞれ送信してこれを記憶させるようにし、上記更新手順を、上記第2の証明鍵証明書をそれぞれの上記クライアントに送信してこれを記憶するよう要求する手順を少なくともそのクライアントの通信相手となる全てのサーバから上記新サーバ証明書を記憶した旨の応答があった後に行うよう定め、さらに、上記第2の証明鍵証明書をそれぞれの上記サーバに送信してこれを記憶するよう要求する動作を少なくともそのサーバの通信相手となる全てのクライアントから上記新クライアント証明書を記憶した旨の応答があった後に行うよう定め、上記各クライアント又は上記各サーバに上記第1の証明鍵証明書を記憶させる際に、その証明書の正当性を従前の証明鍵を用いて確認させ、これが適当なものであると判断した場合にその証明書を記憶させ、上記各クライアント又は上記各サーバに上記第2の証明鍵証明書を記憶させる際に、その証明書の正当性を上記第1の証明鍵証明書に含まれる上記新証明鍵を用いて確認させ、上記第2の証明鍵証明書が適当なものであると判断した場合に、その証明書を記憶させると共に従前の証明鍵証明書及び上記第1の証明鍵証明書を削除させるようにするとよい。

【0037】

さらに、これらのデジタル証明書管理方法において、上記クライアントと上記

サーバとの間の上記相互認証を、SSL又はTLSのプロトコルに従った相互認証とし、上記クライアント証明書及び上記サーバ証明書をそれぞれ上記各クライアント及び上記各サーバの公開鍵証明書とするとよい。

【0038】

また、この発明の更新手順決定方法は、クライアント・サーバシステムを構成する1又は複数のクライアントと1又は複数のサーバとに記憶させ、これらの間で相互認証に使用するデジタル証明書の正当性を確認するための証明鍵を、上記各クライアント及び上記各サーバと通信可能なデジタル証明書管理装置によって更新する際の更新手順を定める更新手順決定方法であって、上記デジタル証明書管理装置が、上記クライアント・サーバシステムを構成する各ノードについて、そのノードの通信相手及びその通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶しておき、その情報をもとに、上記更新手順を、それぞれの上記サーバに、そのサーバが上記相互認証に使用するための、更新用の新証明鍵を用いて正当性を確認可能な新デジタル証明書である上記新サーバ証明書を送信してこれを記憶させる手順を、そのサーバの通信相手となる全てのクライアントから上記新証明鍵を記憶した旨の応答があった後に行うよう定めるものである。

【0039】

また、この発明のプログラムは、クライアント・サーバシステムを構成する1又は複数のクライアント及び1又は複数のサーバと通信可能なデジタル証明書管理装置を制御するコンピュータを、上記各クライアントと上記各サーバとの間で相互認証に使用するデジタル証明書の正当性を確認するための証明鍵を更新する証明鍵更新手段と、上記クライアント・サーバシステムを構成する各ノードについて、そのノードの通信相手及びその通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶する構成記憶手段と、その構成記憶手段に記憶している情報をもとに、上記証明鍵更新手段による証明鍵の更新手順を制御する更新順制御手段として機能させるためのプログラムにおいて、上記証明鍵更新手段に、更新用の新証明鍵を取得する手段と、その新証明鍵を用いて正当性を確認可能な、上記相互認証に使用するための新デジタル証明書を取得する手段と

、上記クライアントのための新デジタル証明書である新クライアント証明書と、上記新証明鍵とをそれぞれ上記クライアントに送信してこれを記憶するよう要求する第1の更新要求手段と、上記サーバのための新デジタル証明書である新サーバ証明書と、上記新証明鍵とをそれぞれ上記サーバに送信してこれを記憶するよう要求する第2の更新要求手段との機能を設け、上記更新順制御手段が、上記第2の更新要求手段がそれぞれの上記サーバに対して上記新サーバ証明書を送信してこれを記憶するよう要求する動作を、そのサーバの通信相手となる全てのクライアントからの上記新証明鍵を記憶した旨の応答があった後に行うように上記更新手順を制御するようにしたものである。

【0040】

このようなプログラムにおいて、上記更新順制御手段が、上記第1の更新要求手段がそれぞれの上記クライアントに上記新クライアント証明書を送信してこれを記憶するよう要求する動作を、そのクライアントの通信相手となる全てのサーバからの上記新証明鍵を記憶した旨の応答があった後に行うように上記更新手順を制御するようにするとよい。

さらに、上記更新順制御手段が、上記第1の更新要求手段が上記新クライアント証明書と上記新証明鍵とを同時に上記各クライアントに送信してこれらを記憶するよう要求し、上記第2の更新要求手段が、それぞれの上記サーバに対して、そのサーバの通信相手となる全てのクライアントからの上記新証明鍵を記憶した旨の応答があった後で、上記新サーバ証明書と上記新証明鍵とを同時に上記サーバに送信し、これらを記憶するよう要求するように上記更新手順を制御するようにするとよい。

【0041】

また、これらのプログラムにおいて、上記コンピュータを、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む証明鍵証明書を取得する手段として機能させるためのプログラムをさらに含め、上記第1の更新要求手段が、上記新証明鍵を上記証明鍵証明書の形式で上記各クライアントに送信してここに含まれる証明鍵を記憶するよう要求するようにし、上記第2の更新要求手段が、上記新証明鍵を上記証明鍵証明書の形式で上記各サーバに送信

してここに含まれる証明鍵を記憶するよう要求するようになるとよい。

【0 0 4 2】

あるいは、上記コンピュータを、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む第1の証明鍵証明書を取得する手段と、上記新証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む第2の証明鍵証明書を取得する手段として機能させるためのプログラムをさらに含め、上記第1の更新要求手段を、上記新証明鍵を上記第1及び第2の証明鍵証明書の形式でそれぞれ上記各クライアントに送信してこれを記憶するよう要求し、上記各クライアントに、上記第2の証明鍵証明書を記憶する場合には従前の証明鍵証明書及び上記第1の証明鍵証明書を削除させる機能を有するようにし、上記第2の更新要求手段を、上記新証明鍵を上記第1及び第2の証明鍵証明書の形式でそれぞれ上記各サーバに送信してこれを記憶するよう要求し、上記各サーバに、上記第2の証明鍵証明書を記憶する場合には従前の証明鍵証明書及び上記第1の証明鍵証明書を削除させる機能を有するようにし、上記更新順制御手段が、上記第1の更新要求手段が上記第2の証明鍵証明書をそれぞれの上記クライアントに送信してこれを記憶するよう要求する動作を、少なくともそのクライアントの通信相手となる全てのサーバから上記新サーバ証明書を記憶した旨の応答があった後に行い、上記第2の更新要求手段が上記第2の証明鍵証明書をそれぞれの上記サーバに送信してこれを記憶するよう要求する動作を、少なくともそのサーバの通信相手となる全てのクライアントから上記新クライアント証明書を記憶した旨の応答があった後に行うように上記更新手順を制御するようになるとよい。

さらに、これらのプログラムにおいて、上記相互認証を、SSL又はTLSのプロトコルに従った相互認証とし、上記クライアント証明書及び上記サーバ証明書をそれぞれ上記各クライアント及び上記各サーバの公開鍵証明書とするとよい。

【0 0 4 3】

【発明の実施の形態】

以下、この発明の好ましい実施の形態を図面を参照して説明する。

〔第 1 の実施形態：図 1 乃至図 1 1〕

まず、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント及びサーバによって構成される、この発明のデジタル証明書管理システムの第 1 の実施形態の構成について説明する。この実施形態においては、各 1 つのクライアント及びサーバによってクライアント・サーバシステムを構成しており、この実施形態は、この発明を最も基本的なシステムに適用した例である。図 2 に、このデジタル証明書管理システムを構成する各装置の、この発明の特徴となる部分の機能構成を示す機能ブロック図を示す。図 2 において、この発明の特徴と関連しない部分の図示は省略している。

【0 0 4 4】

図 2 に示すように、このデジタル証明書管理システムは、証明書管理装置 1 0、サーバ装置 3 0、クライアント装置 4 0 によって構成される。

そして、クライアント装置（クライアント）4 0 及びサーバ装置（サーバ）3 0 は、公開鍵暗号とデジタル証明書を用いる認証方式である SSL による相互認証によって互いを正当な通信相手として認証した場合に、互いに通信を確立させるようにしている。そして、クライアント装置 4 0 が送信した要求に対し、サーバ装置 3 0 が必要な処理を行って応答を返すことにより、クライアント・サーバシステムとして機能する。証明書管理装置 1 0 は、その相互認証に用いるデジタル証明書を発行し、またそのデジタル証明書の管理や更新等を行うための装置であり、CA に相当する。

【0 0 4 5】

なお、実際のシステムにおいては、サーバ装置 3 0 がクライアントの機能を併せ持ったり、クライアント装置 4 0 がサーバの機能を併せ持ったりすることも考えられる。そして、サーバ装置 3 0 がクライアントとして機能して、サーバとして機能するクライアント装置 4 0 に要求を送信することもありうるが、このような場合には、後述する第 2 の実施形態に準ずる動作を行うようにすればよい。従って、ここでは後述するルート鍵更新処理においてサーバとして機能する装置をサーバ装置、クライアントとして機能する装置をクライアント装置と呼ぶものと

する。

【0046】

このようなデジタル証明書管理システムにおいて、上述のクライアント装置 40 からサーバ装置 30 への送信も含め、証明書管理装置 10、サーバ装置 30、クライアント装置 40 の各ノードは、RPC (remote procedure call) により、相互の実装するアプリケーションプログラムのメソッドに対する処理の依頼である「要求」を送信し、この依頼された処理の結果である「応答」を取得することができるようになっている。

【0047】

すなわち、サーバ装置 30 又はクライアント装置 40 では、証明書管理装置 10 への要求を生成してこれを証明書管理装置 10 へ引き渡し、この要求に対する応答を取得できる一方で、証明書管理装置 10 は、クライアント・サーバシステム側への要求を生成してこれをサーバ装置 30 へ引き渡し、この要求に対する応答を取得できるようになっている。この要求には、サーバ装置 30 にクライアント装置 40 に対して各種要求を送信させ、クライアント装置 40 からの応答をサーバ装置 30 を介して取得することも含まれる。

なお、RPC を実現するために、SOAP (Simple Object Access Protocol)、HTTP (Hyper Text Transfer Protocol)、FTP (File Transfer Protocol)、COM (Component Object Model)、CORBA (Common Object Request Broker Architecture) 等の既知のプロトコル (通信規格)、技術、仕様などを利用することができる。

【0048】

この送受信のデータ送受モデルを図 3 の概念図に示す。

(A) は、証明書管理装置 10 でクライアント装置 40 に対する要求が発生したケースである。このケースでは、証明書管理装置 10 が管理装置側要求 a を生成し、これをサーバ装置 30 を経由して受け取ったクライアント装置 40 がこの要求に対する応答 a を返すというモデルになる。なお、(A) では、応答 a だけでなく応答遅延通知 a' を返信するケースが表記されている。これは、クライアント装置 40 が、サーバ装置 30 を経由して管理装置側要求 a を受け取って、当

該要求に対する応答を即座に返せないと判断したときには、応答遅延通知を通知して一旦接続状態を切断し、次回の接続の際に上記要求に対する応答を改めて引き渡す構成としているためである。

なおここでは、サーバ装置 3 0 からクライアント装置 4 0 に対して通信を要求することはできないので、サーバ装置 3 0 からクライアント装置 4 0 に対して送信すべき要求は、クライアント装置 4 0 からサーバ装置 3 0 に対して接続要求があった場合に、これに対する応答として送信することになる。

【 0 0 4 9 】

(B) は、クライアント装置 4 0 で証明書管理装置 1 0 に対する要求が発生したケースである。このケースでは、クライアント装置 4 0 がクライアント装置側要求 b を生成し、これをサーバ装置 3 0 を経由して受け取った証明書管理装置 1 0 が、当該要求に対する応答 b を返すというモデルになっている。なお、(B) のケースでも、応答を即座に返せないときに応答遅延通知 b' を返すことは (A) のケースと同様である。

【 0 0 5 0 】

次に、このデジタル証明書管理システムを構成する各装置の構成と機能についてより詳細に説明する。

図 1 は、図 2 に示した証明書管理装置のハードウェア構成を示すブロック図である。この図に示す通り、証明書管理装置 1 0 は、CPU 1 1, ROM 1 2, RAM 1 3, HDD 1 4, 通信インタフェース (I/F) 1 5 を備え、これらがシステムバス 1 6 によって接続されている。そして、CPU 1 1 が ROM 1 2 や HDD 1 4 に記憶している各種制御プログラムを実行することによってこの証明書管理装置 1 0 の動作を制御し、後述するようにこの発明に係る各手段 (証明鍵更新手段, 構成記憶手段, 更新順制御手段, 第 1 の更新要求手段, 第 2 の更新要求手段, その他の手段) として機能させる。

なお、証明書管理装置 1 0 のハードウェアとしては、適宜公知のコンピュータを採用することができる。もちろん、必要に応じて他のハードウェアを付加してもよい。

【 0 0 5 1 】

クライアント・サーバシステムを構成するクライアント装置及びサーバ装置については、装置の遠隔管理、電子商取引等の目的に応じて種々の構成をとることができる。例えば、遠隔管理の場合には、プリンタ、FAX装置、コピー機、スキャナ、デジタル複合機等の画像処理装置を始め、ネットワーク家電、自動販売機、医療機器、電源装置、空調システム、ガス・水道・電気等の計量システム等の電子装置を被管理装置であるサーバ装置とし、これらの被管理装置から情報を収集したり、コマンドを送って動作させたりするための管理装置をクライアント装置とすることが考えられる。

【0052】

しかし、クライアント装置及びサーバ装置は、少なくともそれぞれCPU、ROM、RAM、ネットワークを介して外部装置と通信するための通信I/F、および認証処理に必要な情報を記憶する記憶手段を備え、CPUがROM等に記憶した所要の制御プログラムを実行することにより、装置をクライアントあるいはサーバとして機能させることができるものとする。

なお、この通信には、有線、無線を問わず、ネットワークを構築可能な各種通信回線（通信経路）を採用することができる。証明書管理装置10との間の通信についても同様である。

【0053】

図2には、上述のように、各装置のこの発明の特徴となる部分の機能構成を示している。

まず、証明書管理装置10は、証明用鍵作成部21、証明書発行部22、証明書管理部23、証明書更新部24、通信機能部25、構成記憶部26、更新順制御部27を備えている。

証明用鍵作成部21は、デジタル署名の作成に用いる証明用私有鍵であるルート私有鍵と、そのデジタル証明書の正当性を確認するための、ルート私有鍵と対応する証明用公開鍵（証明鍵）であるルート鍵とを作成する証明用鍵作成手段の機能を有する。

【0054】

証明書発行部22は、サーバ装置30とクライアント装置40との間の認証処

理に用いる認証情報であるクライアント公開鍵およびサーバ公開鍵にデジタル署名を付して、デジタル証明書であるクライアント公開鍵証明書およびサーバ公開鍵証明書として発行する証明書発行手段の機能を有する。また、クライアント公開鍵、クライアント私有鍵、サーバ公開鍵、サーバ私有鍵の作成及び、ルート鍵にデジタル署名を付したデジタル証明書であるルート鍵証明書の作成も、この証明書発行部 2 2 の機能である。

証明書管理部 2 3 は、証明書発行部 2 2 が発行したデジタル証明書、その作成に用いたルート私有鍵、およびそのルート私有鍵と対応するルート鍵を管理する証明書管理手段の機能を有する。そして、これらの証明書や鍵を、その有効期限や発行先、ID、更新の有無等の情報と共に記憶する。

【0 0 5 5】

証明書更新部 2 4 は、ルート鍵の更新を行う場合に、有効なルート私有鍵の各々について、新たなルート私有鍵（新ルート私有鍵）及びこれと対応する新たなルート鍵（新ルート鍵）を証明用鍵作成部 2 1 に作成させ、これらを更新する証明用鍵更新手段の機能を有する。さらに、この更新に当たって、証明書発行部 2 2 に新ルート私有鍵を用いてデジタル署名を付した新たなクライアント公開鍵証明書（新クライアント公開鍵証明書）、新たなサーバ公開鍵証明書（新サーバ公開鍵証明書）及び新たなルート鍵証明書（新ルート鍵証明書）を発行させ、通信機能部 2 5 によってこれらをサーバ装置 3 0 及びクライアント装置 4 0 に送信させ、サーバ装置 3 0 及びクライアント装置 4 0 にこれらの更新を要求させる機能も有する。また、詳細は後述するが、更新に必要な各処理の手順や進捗状況の管理は、更新順制御部 2 7 が行う。

【0 0 5 6】

通信機能部 2 5 は、ネットワークを介して外部装置と通信する機能を有し、証明書管理部 2 3 の指示に応じて必要なデータをサーバ装置 3 0 及びクライアント装置 4 0 に送信したり、受信したデータを証明書更新部 2 4 に渡したりする。

構成記憶部 2 6 は、証明書管理装置 1 0 がデジタル証明書の管理を行う対象であるクライアント・サーバシステムを構成する各ノード（ここではサーバ装置 3 0 及びクライアント装置 4 0）について、少なくとも該ノードの通信相手及び該

通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶し、構成記憶手段の機能を有する。ここではさらに、各ノードが相互認証に使用する私有鍵、公開鍵証明書、およびルート鍵証明書の ID 及び、鍵や証明書の更新状態に関する情報も記憶するものとする。

更新順制御部 2 7 は、ルート鍵の更新の必要が生じた場合に、構成記憶部 2 6 に記憶している情報をもとに、証明書更新部 2 4 による鍵や証明書の更新手順を定め、証明書更新部 2 4 に更新動作を行わせると共にこれを制御する更新順制御手段として機能する。

そして、これらの各部の機能は、図 1 に示した CPU 1 1 が所要の制御プログラムを実行して証明書管理装置 1 0 の各部の動作を制御することにより実現される。

【 0 0 5 7 】

一方、サーバ装置 3 0 は、証明書記憶部 3 1，通信機能部 3 2，サーバ機能部 3 3 を備えている。

証明書記憶部 3 1 は、SSL による相互認証に用いる鍵を記憶する機能を有し、図 4 6 に示したルート鍵証明書、サーバ私有鍵、およびサーバ公開鍵証明書を記憶する。

通信機能部 3 2 は、ネットワークを介して外部装置と通信する機能を有し、受信したデータをサーバ機能部 3 3 に渡し、またサーバ機能部 3 3 の指示に従ってデータを外部装置に送信する。

【 0 0 5 8 】

サーバ機能部 3 3 は、クライアント装置 4 0 から受信した要求に対して所要の処理を行って応答を返すサーバとしての機能を有する。また、以下に詳述するが、証明書管理装置 1 0 から受信した証明書更新等の要求に対しても、所要の処理を行って応答を返す。

そして、これらの各部の機能は、サーバ装置 3 0 の CPU が所要の制御プログラムを実行してサーバ装置 3 0 の各部の動作を制御することにより実現される。

【 0 0 5 9 】

また、クライアント装置 4 0 は、証明書記憶部 4 1，通信機能部 4 2，クライ

アント機能部 4 3 を備えている。

証明書記憶部 4 1 は、S S L による相互認証に用いる鍵を記憶する機能を有し、図 4 6 に示したルート鍵証明書、クライアント私有鍵、およびクライアント公開鍵証明書を記憶する。

通信機能部 4 2 は、ネットワークを介して外部装置と通信する機能を有し、受信したデータをクライアント機能部 4 3 に渡し、またクライアント機能部 4 3 の指示に従ってデータを外部装置に送信する。

【0 0 6 0】

クライアント機能部 4 3 は、ユーザからの操作、図示しないセンサが検出した状態変化、あるいは図示しないタイマによって計測した所定時間経過等をトリガとして、サーバ装置 3 0 に対して所要の要求を送信し、サーバ装置 3 0 からこれに対する応答を受信した場合にはその内容に従った処理を行うクライアントとしての機能を有する。また、以下に詳述するが、応答として証明書管理装置 1 0 からの証明書更新等の要求を受信した場合には、所要の処理を行って応答を返す。

そして、これらの各部の機能は、クライアント装置 4 0 の C P U が所要の制御プログラムを実行してクライアント装置 4 0 の各部の動作を制御することにより実現される。

【0 0 6 1】

なお、このデジタル証明書管理装置において、証明書管理装置 1 0 が直接通信可能なのは、クライアント・サーバシステムを構成する装置のうちサーバ装置 3 0 のみであり、証明書管理装置 1 0 からクライアント装置 4 0 に対する要求は、サーバ装置 3 0 が中継して送るものとする。クライアント装置 4 0 から証明書管理装置 1 0 への応答も、同様である。

また、上記のサーバ装置 3 0 及びクライアント装置 4 0 には、工場出荷時あるいはそれに順ずる時期、少なくともユーザが相互認証処理の運用を開始する前に、初めのルート鍵を記憶させておくものとする。このとき、公開鍵証明書及び秘密鍵も共に記憶させるようにするとよい。

【0 0 6 2】

次に、このような基本的な機能を有する図 2 に示したデジタル証明書管理シス

テムにおけるこの発明の特徴に関連する処理である、ルート鍵更新処理およびそのために必要な構成について説明する。

なお、以下の説明に用いるシーケンス図に記載するサーバ装置 30 とクライアント装置 40 と間の通信処理に際しては、個々に図示はしていないが、通信の確立前に従来の技術の項で図 46 を用いて説明した SSL による相互認証を行い、認証が成功した場合のみデータの転送を行うものとする。そして、この相互認証処理に支障を来さないようにルート鍵証明書を更新可能であることが、この発明の特徴である。以下の実施形態についても同様である。

またここでは、証明書管理装置 10 とサーバ装置 30 との間の通信は、直通回線等の、安全（データの改竄や盗聴がなされないこと）を確保できる通信経路を介して行うものとする。

【0063】

また、ここで説明するルート鍵更新処理は、この発明のデジタル証明書管理方法の第 1 の実施形態に係る処理であり、図 4 乃至図 10 のシーケンス図に示す処理を、図 11 のフローチャートに示す順番で実行するものである。そこで、まず図 4 乃至図 10 の各シーケンス図に示す処理の内容を説明してから、図 11 を用いてその実行順について説明する。以下の各図に示す処理は、証明書管理装置 10、サーバ装置 30、クライアント装置 40 の各 CPU が、所要の制御プログラムを実行することによって行うものである。

【0064】

まず図 4 のシーケンス図に処理 S としてルート鍵証明書作成処理を示す。

この処理においては、証明書管理装置 10 は、ステップ S101 で、有効なルート私有鍵について、新たなルート私有鍵とルート鍵のペアを作成する。ここで、「有効な」ルート私有鍵とは、その時点でクライアント・サーバシステムにおける相互認証に使用中のルート私有鍵という意味であり、より正確には、そのルート私有鍵を用いてデジタル署名を付した証明書が、認証処理に用いられる状態でサーバ装置 30 又はクライアント装置 40 に記憶されているものをいうものとする。過去に作成した私有鍵が有効か否かは、証明書管理部 23 に記憶している公開鍵証明書及びルート鍵証明書の有効期限やその更新の有無の情報や、構成記

憶部 26 に記憶している各ノードが使用している公開鍵証明書及びルート鍵証明書の ID の情報、および証明書に含まれる、デジタル署名に使用したルート私有鍵の識別情報等の情報を基に判断することができる。また、新たな鍵と置き換えられるべきそれまでの鍵を、「従前の」鍵と呼ぶことにする。証明書についても同様である。

そして、ステップ S102 で、ステップ S101 で作成した新ルート鍵に従前のルート私有鍵を用いたデジタル署名を付し、第 1 の証明鍵証明書である配布用ルート鍵証明書を作成する。

以上がルート鍵証明書作成処理である。

【0065】

次に、図 5 のシーケンス図に処理 1 としてサーバ装置のルート鍵証明書記憶処理を示す。

この処理においては、まずステップ S111 で、証明書管理装置 10 がサーバ装置 30 に対して、図 4 のステップ S102 で作成した配布用ルート鍵証明書と共に、その更新要求を送信する。この処理において、証明書管理装置 10 の CPU11 が第 2 の更新要求手段として機能する。

【0066】

サーバ装置 30 は、この要求を受け取ると、ステップ S112 で従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認する。上述のように、配布用ルート鍵証明書には、従前のルート私有鍵を用いたデジタル署名を付しているため、従前のルート鍵を用いてその内容を復号化し、確かに証明書管理装置 10 によって発行されたものであることを確認できる。また、このとき、従来の技術の項で図 47 を用いて説明したようにルート鍵が損傷や改竄等を受けていないことも確認できる。従って、このような配布用ルート鍵証明書を用いることにより、受け取ったルート鍵の正当性を人手によらず確認できることになる。

そして、これが確認できると、次のステップ S113 で配布用ルート鍵証明書を証明書記憶部 31 に記憶する。このとき、まだ従前のルート鍵証明書は消去しない。従って、証明書記憶部 31 には 2 つのルート鍵証明書が記憶された状態となる。

【 0 0 6 7 】

この状態で認証処理を行う場合、受信した公開鍵証明書の正当性を確認する際には、2つのルート鍵証明書を順次用いて確認を試み、どちらかのルート鍵証明書を用いて確認が成功すれば、正当性が確認できたものとする。従って、新旧どちらのルート私有鍵を用いてデジタル署名を付したデジタル証明書であっても、その正当性を確認することができる。なお、配布用ルート鍵証明書を認証処理に使用する際の、ルート鍵に破損や改竄がないことの確認は、従前のルート鍵証明書を用いて行うことができる。これらのステップ S 1 1 2 及び S 1 1 3 において、サーバ装置 3 0 の C P U が第 2 のサーバ側更新手段として機能する。

サーバ装置 3 0 はその後、ステップ S 1 1 4 で証明書管理装置 1 0 に対して更新要求に対する応答として結果通知を返し、配布用ルート鍵証明書の記憶が成功していればその旨を、何らかの理由で失敗していればその旨を伝える。

以上がサーバ装置のルート鍵証明書記憶処理である。

【 0 0 6 8 】

次に、図 6 のシーケンス図に処理 2 としてクライアント装置のルート鍵証明書記憶処理を示す。

この処理においては、まずステップ S 1 2 1 で、証明書管理装置 1 0 がサーバ装置 3 0 に対して、図 4 のステップ S 1 0 2 で作成した配布用ルート鍵証明書と共に、その更新要求をクライアント装置 4 0 に送信するよう要求する更新要求送信要求を送信する。サーバ装置 3 0 は、これに応じてクライアント装置 4 0 に対して配布用ルート鍵証明書とその更新要求とを送信するのであるが、サーバ装置 3 0 側から送信要求を行うことはできない。そこで、クライアント装置 4 0 が所定のタイミングで定期的にサーバ装置 3 0 に対してポーリングして通信を要求するようにし (S 1 2 2) 、これに対する応答として配布用ルート鍵証明書とその更新要求とを送信するようにしている (S 1 2 3) 。

【 0 0 6 9 】

なお、クライアント装置 4 0 がサーバ装置 3 0 に対するポーリングや接続要求を H T T P リクエストとして送信し、サーバ装置 3 0 からクライアント装置 4 0 に対して送信する要求やデータをこれに対する応答である H T T P レスポンスと

して送信するようにするとよい。このようにすれば、クライアント装置40がファイアウォールの内側に設置されている場合でも、これを越えてサーバ装置30からクライアント装置40にデータを転送することができる。

【0070】

ファイアウォールを越える手段はこれに限られるものではなく、例えば、SMTP (Simple Mail Transfer Protocol) を利用して、送信したいデータを記載あるいは添付したメールを送信することも考えられる。ただし、信頼性の面ではHTTPが優れている。

以上の処理により、証明書管理装置10からクライアント装置40に、サーバ装置30を介して配布用ルート鍵証明書とその更新要求とが送信されることになり、ステップS121の処理においては、証明書管理装置10のCPU11が第1の更新要求手段として機能する。

【0071】

クライアント装置40は、この要求を受け取ると、ステップS124で従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認する。そして、これが確認できると、次のステップS125で配布用ルート鍵証明書を証明書記憶部41に記憶する。このとき、まだ従前のルート鍵証明書は消去しない。これらの確認と記憶の詳細については、図5のステップS112及びS113の場合と同様であり、これらのステップにおいて、クライアント装置40のCPUが第2のクライアント側更新手段として機能する。

クライアント装置40はその後、ステップS126で証明書管理装置10に対して更新要求に対する応答として結果通知を返すが、これはまずサーバ装置30に対して送信し、サーバ装置30がステップS127で証明書管理装置に対して送信する。

以上がクライアント装置のルート鍵証明書記憶処理である。

【0072】

次に、図7のシーケンス図に処理3としてクライアント装置の公開鍵証明書記憶処理を示す。

この処理においてはまずステップS131で、証明書管理装置10が、クライ

アント装置 40 に対して発行してあるクライアント公開鍵に、新ルート私有鍵を用いたデジタル署名を付して新クライアント公開鍵証明書を作成する。なお、クライアント私有鍵は更新しないので、クライアント公開鍵自体も更新する必要はない。

【0073】

そしてステップ S 132 で、証明書管理装置 10 がサーバ装置 30 に対して、ステップ S 131 で作成した新クライアント公開鍵証明書と共に、その更新要求をクライアント装置 40 に送信するよう要求する更新要求送信要求を送信する。サーバ装置 30 は、これに応じて、図 6 のステップ S 122 及び S 123 の場合と同様に、クライアント装置 40 からのポーリング (S 133) に対する応答として新クライアント公開鍵証明書とその更新要求とを送信するようにしている (S 134)。

以上の処理により、証明書管理装置 10 からクライアント装置 40 にサーバ装置 30 を介して新クライアント公開鍵証明書とその更新要求とが送信されることになり、ステップ S 132 の処理においては、証明書管理装置 10 の CPU 11 が第 1 の更新要求手段として機能する。

【0074】

クライアント装置 40 は、この要求を受け取るとステップ S 135 で、図 6 のステップ S 125 で記憶した配布用ルート鍵証明書を用いて新クライアント公開鍵証明書の正当性を確認する。上述のように、新クライアント公開鍵証明書には、新ルート私有鍵を用いたデジタル署名を付しているもので、配布用ルート鍵証明書に含まれる新ルート鍵を用いてその内容を復号化し、確かに証明書管理装置 10 によってクライアント装置 40 に対して発行されたものであることを確認できる。そして、これが確認できると、次のステップ S 136 で新クライアント公開鍵証明書を証明書記憶部 41 に記憶する。これらのステップ S 135 及び S 136 において、クライアント装置 40 の CPU が第 1 のクライアント側更新手段として機能する。

【0075】

このとき、まだ従前のクライアント公開鍵証明書は消去しない。従って、証明

書記憶部 4 1 には 2 つのクライアント公開鍵証明書が記憶された状態となる。この状態で認証処理を行い、通信相手に対して公開鍵証明書を送信する場合には、まず新公開鍵証明書を送信するものとする。

この場合、通信相手が既に新ルート鍵を（配布用ルート鍵証明書又は後述する新ルート鍵証明書として）記憶していれば、新公開鍵証明書のデジタル署名を復号化できるので、問題なく認証を受けることができる。一方、通信相手がまた新ルート鍵を記憶していない場合には、新公開鍵証明書のデジタル署名を復号化できず、認証が失敗した旨の応答を受けることになる。しかしこの場合でも、再度通信を要求し、この際に従前の公開鍵証明書を送信するようにすれば、従前のルート鍵によってそこに付されたデジタル署名を復号化できるので、問題なく認証を受けることができる。

【 0 0 7 6 】

従って、2 つの公開鍵証明書を記憶しておけば、通信相手が新ルート鍵を記憶していない場合に多少のオーバーヘッドが生じることはあるが、問題なく相互認証を行うことができる。なお、2 つの公開鍵証明書に含まれる公開鍵本体は同じものであるので、クライアント私有鍵を用いて暗号化したデータの復号化は、どちらの公開鍵証明書を用いた場合でも同じように行うことができる。

クライアント装置 4 0 はその後、ステップ S 1 3 7 で証明書管理装置 1 0 に対して更新要求に対する応答として結果通知を返すが、これはまずサーバ装置 3 0 に対して送信し、サーバ装置 3 0 がステップ S 1 3 8 で証明書管理装置に対して送信する。

以上がクライアント装置の公開鍵証明書記憶処理である。

【 0 0 7 7 】

次に、図 8 のシーケンス図に処理 4 としてサーバ装置の公開鍵証明書記憶処理を示す。

この処理においてはまずステップ S 1 4 1 で、証明書管理装置 1 0 が、クライアント装置 4 0 に対して発行してあるサーバ公開鍵に、新ルート私有鍵を用いたデジタル署名を付して新サーバ公開鍵証明書を作成する。サーバ公開鍵自体の更新が不要であることは、上述のクライアント公開鍵の場合と同様である。

そしてステップ S 1 4 2 で、証明書管理装置 1 0 がサーバ装置 3 0 に対して、新サーバ公開鍵証明書と共にその更新要求を送信する。この処理において、証明書管理装置 1 0 の CPU 1 1 が第 2 の更新要求手段として機能する。

【 0 0 7 8 】

サーバ装置 3 0 は、この要求を受け取るとステップ S 1 4 3 で、図 5 のステップ S 1 1 3 で記憶した配布用ルート鍵証明書を用いて新公開鍵証明書の正当性を確認する。この点については、図 7 のステップ S 1 3 5 の場合と同様である。そして、これが確認できると、次のステップ S 1 4 4 で新サーバ公開鍵証明書を証明書記憶部 4 1 に記憶し、従前のサーバ公開鍵証明書と置き換える。これらのステップ S 1 4 3 及び S 1 4 4 において、サーバ装置 3 0 の CPU が第 1 のサーバ側更新手段として機能する。

【 0 0 7 9 】

ところで、サーバ装置 3 0 の場合には、クライアント装置 4 0 の場合と異なり、新公開鍵証明書を記憶させる場合に従前のものに追加するのではなくこれと置き換える必要があるのであるが、ここでこの点について説明する。

サーバ装置 3 0 の場合には、クライアント装置 4 0 から接続要求があった場合に公開鍵証明書をクライアント装置 4 0 に送信するのであるが、サーバ公開鍵証明書を複数記憶していたとすると、送信毎にそのうちいずれかを選択して送信することになる。そして、クライアント装置 4 0 側でデジタル証明書を復号化できないようなサーバ公開鍵証明書を送信してしまった場合には、認証は失敗することになる。例えば、クライアント装置 4 0 が新ルート鍵を記憶する前に新サーバ公開鍵証明書を送信した場合等である。

【 0 0 8 0 】

たとえ失敗したとしても、次に接続要求があった場合にもう一方のサーバ公開鍵証明書を送信すればよいという考え方もあるが、不特定多数のクライアント装置から任意のタイミングで接続要求を受け得るサーバ装置の場合、クライアント装置毎に送信すべきサーバ公開鍵証明書を選択することは、現実的ではない。また、クライアント装置がどのような装置であるかは、サーバ装置側では認証が済むまで通常わからないので、最初に送信するサーバ公開鍵証明書を適切に選択す

ることも困難である。従って、サーバ装置にはサーバ公開鍵証明書を1つだけ記憶させ、クライアント装置から接続要求を受けた場合には常にこれを送信するようにする必要があるのである。

【0081】

従って、サーバ装置30では新サーバ公開鍵証明書を記憶させた時点で従前のサーバ公開鍵証明書は削除してしまうので、クライアント装置40に新ルート鍵を記憶させる前にこれを行ってしまうと、クライアント装置側でサーバ公開鍵証明書のデジタル署名を復号化できなくなり、相互認証を行えなくなってしまう。そこで、サーバ装置30の公開鍵証明書記憶処理は、クライアント装置のルート鍵証明書記憶処理の完了後に行う必要がある。

以上のようなステップS144の終了後、サーバ装置30はステップS145で証明書管理装置10に対して更新要求に対する応答として結果通知を返し、新サーバ公開鍵証明書の記憶が成功していればその旨を、何らかの理由で失敗していればその旨を伝える。

以上がサーバ装置の公開鍵証明書記憶処理である。

【0082】

次に、図9のシーケンス図に処理5としてサーバ装置のルート鍵証明書書き換え処理を示す。

この処理においてはまずステップS151で、証明書管理装置10が、新ルート鍵に新ルート私有鍵を用いたデジタル署名を付して第2の証明鍵証明書として新ルート鍵証明書を作成する。そして、ステップS152で証明書管理装置10がサーバ装置30に対して、新ルート鍵証明書と共にその更新要求を送信する。この処理においても、証明書管理装置10のCPU11が第2の更新要求手段として機能する。

【0083】

サーバ装置30は、この要求を受け取ると、ステップS153で配布用ルート鍵証明書を用いて新ルート鍵証明書の正当性を確認する。上述のように、新ルート鍵証明書には、新ルート私有鍵を用いたデジタル署名を付しているなので、配布用ルート鍵証明書に含まれる新ルート鍵を用いてその内容を復号化し、確かに証

明書管理装置 10 によって発行されたものであることを確認できる。

そして、これが確認できると、次のステップ S 154 で新ルート鍵証明書を証明書記憶部 31 に記憶する。そして、配布用ルート鍵証明書及び従前のルート鍵証明書を削除して廃棄し、ルート鍵証明書を新たなものに書き換えてしまう。このようにすると、従前のルート私有鍵を用いてデジタル署名を付したデジタル証明書は復号化できなくなってしまうが、クライアント装置 40 に新クライアント公開鍵証明書を記憶させた後であれば、クライアント装置 40 から送られてくる公開鍵証明書の確認には支障がないので、認証処理に支障を来すことはない。

【0084】

サーバ装置 30 はその後、ステップ S 155 で証明書管理装置 10 に対して更新要求に対する応答として結果通知を返し、新ルート鍵証明書の記憶が成功していればその旨を、何らかの理由で失敗していればその旨を伝える。

以上がサーバ装置のルート鍵証明書書き換え処理である。

【0085】

次に、図 10 のシーケンス図に処理 6 としてクライアント装置のルート鍵証明書書き換え処理を示す。

この処理においてはまずステップ S 161 で、証明書管理装置 10 が、新ルート鍵に新ルート私有鍵を用いたデジタル署名を付して第 2 の証明鍵証明書として新ルート鍵証明書を作成する。これは、図 9 のステップ S 151 で作成するものと同じであるので、ここで作成したものを流用してもよい。逆に図 9 のステップ S 151 で、このステップ S 161 で作成したものを流用してもよい。

【0086】

そしてステップ S 162 で、証明書管理装置 10 がサーバ装置 30 に対して、ステップ S 161 で作成した新ルート鍵証明書と共に、その更新要求をクライアント装置 40 に送信するよう要求する更新要求送信要求を送信する。サーバ装置 30 は、これに応じて、図 6 のステップ S 122 及び S 123 の場合と同様に、クライアント装置 40 からのポーリング (S 163) に対する応答として新ルート鍵証明書とその更新要求とを送信するようにしている (S 164)。

以上の処理により、証明書管理装置 10 からクライアント装置 40 にサーバ装

置 30 を介して新ルート鍵証明書とその更新要求とが送信されることになり、ステップ S 162 の処理においても、証明書管理装置 10 の CPU 11 が第 1 の更新要求手段として機能する。

【0087】

クライアント装置 40 は、この要求を受け取ると、ステップ S 165 で配布用ルート鍵証明書を用いて新ルート鍵証明書の正当性を確認する。そして、これが確認できると、次のステップ S 166 で新ルート鍵証明書を証明書記憶部 41 に記憶する。そして、配布用ルート鍵証明書及び従前のルート鍵証明書を削除して廃棄し、ルート鍵証明書を新たなものに書き換えてしまう。これらの処理については、図 9 のステップ S 153 及び S 154 の場合と同様である。ただし、クライアント装置 40 への新クライアント公開鍵証明書の記憶が済んでいれば、ステップ S 166 で従前のクライアント公開鍵証明書も同時に廃棄してしまってよい。

【0088】

クライアント装置 40 はその後、ステップ S 167 で証明書管理装置 10 に対して更新要求に対する応答として結果通知を返すが、これはまずサーバ装置 30 に対して送信し、サーバ装置 30 がステップ S 168 で証明書管理装置に対して送信する。

以上がクライアント装置のルート鍵証明書書き換え処理である。

【0089】

以上の図 4 乃至図 10 に示した各処理の実行タイミングは、証明書管理装置 10 の更新順制御部 27 が構成記憶部 26 に記憶している情報をもとに更新手順を作成して管理する。そして、その更新手順はここでは図 11 に示すフローチャートのようなものになる。すなわち、ルート鍵の更新事由を検出した場合に、図 11 のフローチャートに示す処理を開始し、まず図 4 に示した処理 S を実行し、その後処理 1 乃至処理 6 を実行する。なお、ルート鍵の更新事由としては、所定の有効期限の到来、管理者の指示等が考えられる。管理者が更新の指示を行う場合としては、ルート私有鍵の第 3 者への漏洩が判明した場合等が考えられる。

また、図 11 において、矢印の先の処理は、矢印の根元側の処理が全て完了し

てから開始する。破線で示した矢印については、その条件は必須ではないが考慮した方が好ましいということを示す。

【0 0 9 0】

具体的には、処理 1 及び処理 2 は処理 S の完了後に開始する。処理 3 は、処理 2 の完了後に開始するが、処理 1 も完了した後に開始する方が好ましい。処理 4 は、処理 1 及び処理 2 の完了後に開始する。処理 5 は、処理 1 及び処理 3 の完了後に開始する。処理 6 は、処理 2 及び処理 4 の完了後に開始する。そして、処理 3 乃至 6 が全て完了した時点で、ルート鍵及び公開鍵証明書の更新が終了したことになる。

【0 0 9 1】

なお。各処理は、更新要求に対する更新成功の応答を受け取った場合に完了したものとすることができる。更新失敗の応答を受け取った場合や処理がタイムアウトした場合には、再度同じ処理を試みるとよいが、所定回数続けて失敗した場合には更新処理全体が失敗したものとするとよい。ルート鍵更新処理を図 1 1 に示す手順で行う場合、サーバ装置 3 0 とクライアント装置 4 0 とは処理のどの時点であっても互いに S S L による相互認証を行うことができるので、このように更新処理が途中で中断してしまっても、サーバ装置 3 0 とクライアント装置 4 0 との間の通信に大きな支障はない。従って、更新処理が失敗した場合に時間をかけて失敗の原因を特定した上で改めて更新処理を行っても、特に大きな問題はない。以後の各実施形態についても同様である。

【0 0 9 2】

このデジタル証明書管理システムにおいては、ルート鍵更新処理をこのような手順で行うことにより、サーバ装置 3 0 とクライアント装置 4 0 との間の相互認証処理に大きな影響を与えることなく、ルート鍵を自動制御で更新することができる。従って、このようなデジタル証明書管理システムを用いることにより、クライアント・サーバシステムにおける S S L による相互認証を、低コストで運用することができる。

証明書管理装置 1 0 とサーバ装置 3 0 との間には、これとは別の安全な通信経路を設ける必要があるが、証明書管理装置 1 0 と 1 つの装置のみとの間に設けれ

ばよいので、特に大きな負担にはならない。証明書管理装置 1 0 とサーバ装置 3 0 とが物理的に近接している場合には専用ケーブルで結ぶ等してこのような経路を設けることは容易であり、この実施形態はこのような場合に好ましいものであると言える。

【0 0 9 3】

図 1 1 に示す処理手順において、この発明の特徴となるのは、まず、処理 4（サーバ装置の公開鍵証明書記憶処理）を処理 2（クライアント装置のルート鍵証明書記憶処理）の後で、すなわちクライアント装置 4 0 から配布用ルート鍵証明書を記憶した旨の応答があった後で実行する点である。

処理 4 の説明において上述したように、サーバ装置 3 0 については公開鍵証明書を同時に 2 つ記憶させると不都合が生じるので、新サーバ公開鍵証明書を記憶させる際には従前のものを廃棄する必要があるのであるが、このような書き換えを行ってしまっても、クライアント装置 4 0 に新ルート鍵を記憶させた後であれば、認証処理に支障が生じることがない。

【0 0 9 4】

また、処理 3（クライアント装置の公開鍵証明書記憶処理）を処理 1（サーバ装置のルート鍵証明書記憶処理）の後で、すなわちサーバ装置 3 0 から配布用ルート鍵証明書を記憶した旨の応答があった後で実行するようにするとよい。

処理 3 の説明で上述したように、クライアント装置 4 0 に新クライアント公開鍵証明書を記憶させた時点でサーバ装置 3 0 に新ルート鍵が記憶されていないと、サーバ装置 3 0 に新ルート鍵が記憶されるまで通信にオーバーヘッドが生じ、効率が悪くなってしまうためである。

【0 0 9 5】

処理 5 と処理 6 については、これらは必須の処理ではないが、従前のルート鍵証明書や公開鍵証明書をいつまでも記憶させておくとすると、記憶容量を無駄に消費することになる。鍵や証明書の記憶には、信頼性の高い記憶手段を用いることが好ましく、従って容量当たりのコストが高いため、この点は大きな問題になる。また、配布用ルート鍵証明書は、自己署名形式でないため、使用する際に従前のルート鍵証明書を参照する必要があり、処理効率が悪い。そこで、処理 5 と

処理 6 を行って、ルート鍵証明書を自己署名形式のものにすると共に、従前の証明書を廃棄するようにするとよい。

【0 0 9 6】

ルート鍵証明書を自己署名形式のものに書き換えるだけであれば、配布用ルート鍵証明書を記憶させた直後に、例えば処理 5 の場合には処理 1 の完了直後に行ってもよいのであるが、この時点では必ずしも従前のルート鍵証明書を廃棄できない。そして、この削除タイミングはサーバ装置 3 0 側では決定することができないので、処理 3 の終了後に再度従前のルート鍵証明書を廃棄する要求を行う必要が生じてしまう。従って、処理 1 と処理 3 の完了後に処理 5 を行うことが、処理の簡略化の点から好ましい。処理 6 についても、同様の理由から処理 2 と処理 4 の完了後に行うことが好ましい。

【0 0 9 7】

なお、ルート鍵は一旦記憶してしまえば一般に外部に送信する必要はないので、その後の破損や改竄は考えにくいことから、ルート鍵証明書ではなく、鍵部分のみを記憶することも考えられる。このような場合には、配布用ルート鍵証明書に含まれる新ルート鍵を記憶してしまえばよいので、証明書管理装置 1 0 から新ルート鍵証明書を別途送信する必要はない。そこで、このような場合、処理 5、処理 6 においては、新ルート鍵証明書を送信せず、従前のルート鍵の廃棄のみを要求するようにすればよい。また、ルート鍵を使用する際に、デジタル署名の確認を行わないようにする場合についても同様である。

なお、この実施形態において、サーバ装置 3 0 からクライアント装置 4 0 への送信は、クライアント装置 4 0 からのポーリングに対する応答として行う例について説明したが、サーバ装置 3 0 がクライアントとしても機能できるようにし、クライアント装置 4 0 がサーバとしても機能できるようにし、これらの機能によって、サーバ装置 3 0 からクライアント装置 4 0 へデータや要求を直接送信できるようにしてもよい。このような場合は、クライアント装置 4 0 によるポーリングは不要である。この点は、以下の実施形態においても同様である。

【0 0 9 8】

〔第 2 の実施形態：図 1 2 乃至図 1 9〕

次に、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント装置及びサーバ装置とによって構成される、この発明のデジタル証明書管理システムの第 2 の実施形態の構成について説明する。この実施形態においても、各 1 つのクライアント及びサーバによってクライアント・サーバシステムを構成しており、この実施形態は、この発明を最も基本的なシステムに適用した第 1 の実施形態とは別の例である。

このデジタル証明書管理システムを構成する各装置の、この発明の特徴となる部分の機能構成を、図 2 と対応する図 1 2 の機能ブロック図に示す。この図において、図 2 と対応する部分には同一の符号を付している。

【0 0 9 9】

この図からわかるように、このデジタル証明書管理システムにおいてはまず、証明書管理装置 1 0 をクライアント・サーバシステムを構成する装置のうちクライアント装置 4 0 のみと直接通信可能とし、証明書管理装置 1 0 からサーバ装置 3 0 に対する要求は、クライアント装置 4 0 が中継して送るものとした点が第 1 の実施形態と異なる。

また、クライアント装置 4 0 にもサーバ機能部 4 4 を設けた点も、第 1 の実施形態の場合と異なるが、このサーバ機能部 4 4 は、受信した要求に対して所要の処理を行って応答を返すサーバとしての機能を有し、証明書管理装置 1 0 との通信のために設けたものである。クライアント装置 4 0 がクライアント機能部 4 3 しか有しないとすると、証明書管理装置 1 0 からクライアント装置 4 0 にデータや要求等を送信する場合に、クライアント装置 4 0 からのポーリングを待つ必要が生じてしまう。

【0 1 0 0】

しかし、ルート鍵の更新処理は頻繁に行われるものではなく、例えば年に 1 回程度であるので、このためにクライアント装置 4 0 が証明書管理装置 1 0 に対して定期的にポーリングを行うとすると、ほとんどの通信が無駄になることになる。そこで、クライアント装置 4 0 にサーバ機能部 4 4 を設け、証明書管理装置 1 0 側から通信を要求できるようにしたものである。このサーバ機能部 4 4 の機能も、クライアント装置 4 0 の CPU が所要の制御プログラムを実行してクライア

ント装置 40 の各部の動作を制御することにより実現されるものである。

【0101】

ただし、クライアント・サーバシステムを構成するサーバ装置 30 との関係においては、クライアント装置 40 は常にクライアントとして機能する。従って、証明書管理装置 10 からサーバ装置 30 への通信を仲介する場合には、通信機能部 42 が証明書管理装置 10 から受信したデータや要求を、サーバ機能部 44 が受け取り、これをクライアント機能部 43 に渡して、クライアント機能部 43 の指示に基づいてサーバ装置 30 に対する通信を要求してサーバ装置 30 に送信することになる。サーバ装置 30 からの応答を証明書管理装置 10 に返す場合には、この逆の処理となる。

【0102】

これらの変更に伴ってルート鍵更新処理のシーケンスは変更されるが、それ以外の点については第 1 の実施形態と同様であるので、説明を省略する。

なおここでも、証明書管理装置 10 とクライアント装置 40 との間の通信は、直通回線等の、安全を確保できる通信経路を介して行うものとする。ただし、この実施形態の場合には、証明書管理装置 10 とクライアント装置 40 との間の通信に SSL を用いることも可能であるが、この場合の構成については変形例として後述する。

【0103】

このデジタル証明書管理システムにおけるルート鍵更新動作は、この発明のデジタル証明書管理方法の第 2 の実施形態に係る動作であり、図 13 乃至図 18 のシーケンス図に示す処理及び図 4 を用いて上述した処理 S を、図 19 のフローチャートに示す順番で実行するものである。そこで、まず図 13 乃至図 18 の各シーケンス図に示す処理の内容を説明してから、図 18 を用いてその実行順について説明する。以下の各図に示す処理は、証明書管理装置 10、サーバ装置 30、クライアント装置 40 の各 CPU が、所要の制御プログラムを実行することによって行うものである。

【0104】

まず、図 13 のシーケンス図に処理 11 としてサーバ装置のルート鍵証明書記

憶処理を示す。

この処理は、図5に示した処理1と同じ目的の処理であるが、ここでは証明書管理装置10と直接通信する装置がクライアント装置40であるため、手順が若干異なるものとなっている。

【0105】

すなわち、まずステップS211で、証明書管理装置10がクライアント装置40に対して、図4のステップS102で作成した配布用ルート鍵証明書と共に、その更新要求をサーバ装置30に送信するよう要求する更新要求送信要求を送信する。そしてクライアント装置40は、これに応じてサーバ装置30に対して配布用ルート鍵証明書とその更新要求とを送信する（S212）。クライアント装置40はサーバ装置30に対して通信を要求できるので、図5の場合のようにポーリングを待つ必要はない。

以上の処理により、証明書管理装置10からサーバ装置30にクライアント装置40を介して配布用ルート鍵証明書とその更新要求とが送信されることになり、ステップS211の処理においては、証明書管理装置10のCPU11が第2の更新要求手段として機能する。

【0106】

サーバ装置30は、ステップS212で送信されてきた更新要求を受け取ると、ステップS213で従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認し、これが確認できると、次のステップS214で配布用ルート鍵証明書を証明書記憶部31に記憶する。これらの処理は、図5のステップS112及びS113の処理と全く同じである。

【0107】

サーバ装置30はその後、ステップS215で証明書管理装置10に対して更新要求に対する応答として結果通知を返すが、これはまずクライアント装置40に対して送信し、クライアント装置40がステップS216で証明書管理装置に対して送信する。なお、この結果通知は、クライアント装置40から受信した更新要求に対する応答として送信することができるので、クライアント装置40からのポーリングを待つ必要はない。

以上がこの実施形態におけるサーバ装置のルート鍵証明書記憶処理である。

【0 1 0 8】

次に、図 1 4 のシーケンス図に処理 1 2 としてクライアント装置のルート鍵証明書記憶処理を示す。

この処理は、図 6 に示した処理 2 と同じ目的の処理であるが、処理 1 1 の場合と同様に手順が若干異なるものとなっている。

この処理においては、まずステップ S 2 2 1 で、証明書管理装置 1 0 がクライアント装置 4 0 に対して、図 4 のステップ S 1 0 2 で作成した配布用ルート鍵証明書とその更新要求を送信する。この処理において、証明書管理装置 1 0 の CPU 1 1 が第 1 の更新要求手段として機能する。

【0 1 0 9】

クライアント装置 4 0 は、この要求を受け取ると、ステップ S 1 2 4 で従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認し、これが確認できると、次のステップ S 1 2 5 で配布用ルート鍵証明書を証明書記憶部 4 1 に記憶する。これらの処理は、図 6 のステップ S 1 2 4 及び S 1 2 5 の処理と全く同じである。

クライアント装置 4 0 はその後、ステップ S 2 2 4 で証明書管理装置 1 0 に対して更新要求に対する応答として結果通知を返す。

以上がこの実施形態におけるクライアント装置のルート鍵証明書記憶処理である。

【0 1 1 0】

以下、図 1 5 に処理 1 3 としてクライアント装置の公開鍵証明書記憶処理を、図 1 6 に処理 1 4 としてサーバ装置の公開鍵証明書記憶処理を、図 1 7 に処理 1 5 としてサーバ装置のルート鍵証明書書き換え処理を、図 1 8 に処理 1 6 としてクライアント装置のルート鍵証明書書き換え処理をそれぞれ示すが、これらは、第 1 の実施形態で図 7 乃至図 1 0 を用いてそれぞれ説明した処理 3 乃至処理 6 と同じ目的の処理であり、証明書管理装置 1 0 と直接通信する装置がクライアント装置 4 0 であることに伴って、処理 1 1 及び処理 1 2 の場合と同様に通信手順を若干変更したのみである。そこで、これらの処理についての説明は省略する。

【0111】

また、以上の図13乃至図18に示した各処理及び図4に示した処理Sの実行タイミングは、証明書管理装置10の更新順制御部27が構成記憶部26に記憶している情報をもとに更新手順を作成して管理する。そして、その更新手順はここでは図19に示すフローチャートのようなものになる。すなわち、ルート鍵の更新を行う場合には、まず図4に示した処理Sを実行し、その後処理11乃至処理16を実行する。

図19の記載から明らかなように、この第2の実施形態におけるルート鍵更新処理は、図11に示した第1の実施形態の場合と対応する処理を、同様な順序で行うものである。そして、このことによる効果も、第1の実施形態の場合と同様である。

【0112】

すなわち、この第2の実施形態のデジタル証明書管理システムにおいては、ルート鍵更新処理をこのような手順で行うことにより、証明書管理装置10がクライアント・サーバシステムを構成する装置のうちクライアント装置40のみと通信可能な場合でも、第1の実施形態の場合と同様に、サーバ装置30とクライアント装置40との間の相互認証処理に大きな影響を与えることなくルート鍵を自動制御で更新することができる。従って、このようなデジタル証明書管理システムを用いることにより、クライアント・サーバシステムにおけるSSLによる相互認証を、低コストで運用することができる。

また、この実施形態においては、クライアント装置40にサーバ機能部44を設ける必要はあるが、ルート鍵更新処理の手順にポーリング待ちを必要とする箇所がないため、処理を速やかに進め、短期間で完了させることができる。

【0113】

〔第3の実施形態：図20乃至図23〕

次に、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント装置及びサーバ装置とによって構成される、この発明のデジタル証明書管理システムの第3の実施形態の構成について説明する。

このデジタル証明書管理システムは、ルート鍵更新処理の内容が第1の実施形態のデジタル証明書管理システムと異なるのみであり、装置の構成は第1の実施形態のものと同様であるのでその説明は省略する。

【0114】

このデジタル証明書管理システムにおけるルート鍵更新動作は、この発明のデジタル証明書管理方法の第3の実施形態に係る動作であり、図20乃至図23のシーケンス図に示す処理を、この順で実行するものである。以下の各図に示す処理は、証明書管理装置10、サーバ装置30、クライアント装置40の各CPUが、所要の制御プログラムを実行することによって行うものである。

このデジタル証明書管理システムの証明書管理装置10は、ルート鍵の更新事由を検出すると、図20のシーケンス図に示す処理を開始する。

【0115】

図20に示す処理は、第1の実施形態の説明において図4に示した処理Sと対応する処理Tである。そして、まずステップS301及びS302において、図4のステップS101及びS102の場合と同様に、有効なルート私有鍵について、新たなルート私有鍵とルート鍵のペアを作成すると共に、その新ルート鍵に従前のルート私有鍵を用いたデジタル署名を付し、第1の証明鍵証明書である配布用ルート鍵証明書を作成する。

そしてさらに、ステップS303において、図9のステップS151の場合と同様に、新ルート鍵に新ルート私有鍵を用いたデジタル署名を付して第2の証明鍵証明書として新ルート鍵証明書を作成する。

【0116】

その後、続いて図21のシーケンス図に示す処理21を行う。この処理は、第1の実施形態の説明において図6に示した処理2及び図7に示した処理3を併せ、さらに図10に示した処理6の一部を加えた処理に相当する。

ここではまず、ステップS311で、図7のステップS131の場合と同様に、証明書管理装置10がクライアント公開鍵に新ルート私有鍵を用いたデジタル署名を付して新クライアント公開鍵証明書を作成する。

【0117】

そしてステップ S 3 1 2 で、証明書管理装置 1 0 がサーバ装置 3 0 に対して、図 2 0 のステップ S 3 0 2 で作成した配布用ルート鍵証明書と、図 2 0 のステップ S 3 0 3 で作成した新ルート鍵証明書と、ステップ S 3 1 1 で作成した新クライアント公開鍵証明書と共に、これらについての更新要求をクライアント装置 4 0 に送信するよう要求する更新要求送信要求を送信する。サーバ装置 3 0 はこれに応じて、図 6 のステップ S 1 2 2 及び S 1 2 3 の場合と同様に、クライアント装置 4 0 からのポーリング (S 3 1 3) に対する応答としてこれらの証明書とそれらについての更新要求とを送信するようにしている (S 3 1 4)。

以上の処理により、証明書管理装置 1 0 からクライアント装置 4 0 にサーバ装置 3 0 を介して上記の各証明書とそれらについての更新要求とが送信されることになり、ステップ S 3 1 2 の処理においては、証明書管理装置 1 0 の CPU 1 1 が第 1 の更新要求手段として機能する。

【0118】

クライアント装置 4 0 は、この要求を受け取ると、ステップ S 3 1 5 及び S 3 1 6 で、図 6 のステップ S 1 2 4 及び S 1 2 5 の場合と同様に、従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認し、これが確認できると配布用ルート鍵証明書を証明書記憶部 4 1 に記憶する。このとき、まだ従前のルート鍵証明書は消去しない。

そしてさらにステップ S 3 1 7 で、図 1 0 のステップ S 1 6 5 の場合と同様に、記憶した配布用ルート鍵証明書を用いて新ルート鍵証明書の正当性を確認する。そして、これが確認できると、次のステップ S 3 1 8 で新ルート鍵証明書を証明書記憶部 4 1 に記憶する。この時点で配布用ルート鍵は消去してしまってもよいが、ここでは記憶したままとする。

これらのステップ S 3 1 5 乃至 S 3 1 8 の処理において、クライアント装置 4 0 の CPU が第 2 のクライアント側更新手段として機能する。

【0119】

次に、ステップ S 3 1 9 及び S 3 2 0 で、図 7 のステップ S 1 3 5 及び S 1 3 6 の場合と同様に、新クライアント公開鍵証明書の正当性を確認し、これが確認できると、新クライアント公開鍵証明書を証明書記憶部 4 1 に記憶する。ただし

、ここでは既に新ルート鍵証明書を記憶しているので、新クライアント公開鍵証明書の正当性は、配布用ルート鍵証明書ではなく新ルート鍵証明書を用いて行うことができる。これらのステップ S 3 1 9 及び S 3 2 0 において、クライアント装置 4 0 の C P U が第 1 のクライアント側更新手段として機能する。

【0 1 2 0】

このとき、まだ従前のクライアント公開鍵証明書は消去しない。従って、証明書記憶部 4 1 には 2 つのクライアント公開鍵証明書が記憶された状態となる。この状態で通信相手に対して公開鍵証明書を送信する場合には、まず新公開鍵証明書を送信するものとする。ここではまだサーバ装置 3 0 に新ルート鍵を記憶させていないので、サーバ装置 3 0 は新公開鍵証明書のデジタル署名を復号化できず、認証が失敗した旨の応答を受けることになる。しかしこの場合でも、再度通信を要求し、この際に従前の公開鍵証明書を送信すれば、従前のルート鍵によってそこに付されたデジタル署名を復号化できるので、問題なく認証を受けることができる。

【0 1 2 1】

なお、ステップ S 3 1 9 及び S 3 2 0 の処理を、ステップ S 3 1 7 及び S 3 1 8 の処理より前に行うようにしてもよい。この場合には、ステップ S 3 1 9 における正当性の確認は、配布用ルート鍵証明書を用いて行うことになる。

クライアント装置 4 0 はその後、ステップ S 3 2 1 で、証明書管理装置 1 0 に対して更新要求に対する応答として結果通知を返すが、これはまずサーバ装置 3 0 に対して送信し、サーバ装置 3 0 がステップ S 3 2 2 で証明書管理装置 1 0 に対して送信する。

【0 1 2 2】

その後、続いて図 2 2 のシーケンス図に示す処理 2 2 を行う。この処理は、第 1 の実施形態の説明において図 5 に示した処理 1 及び図 8 に示した処理 4 を併せ、さらに図 9 に示した処理 5 の一部を加えた処理に相当する。

ここではまず、ステップ S 3 2 3 で、図 8 のステップ S 1 4 1 の場合と同様に、証明書管理装置 1 0 がサーバ公開鍵に新ルート私有鍵を用いたデジタル署名を付して新サーバ公開鍵証明書を作成する。

【0123】

そして、ステップS324で、証明書管理装置10がサーバ装置30に対して、図20のステップS302で作成した配布用ルート鍵証明書と、図20のステップS303で作成した新ルート鍵証明書と、ステップS323で作成した新サーバ公開鍵証明書と共に、これらについての更新要求を送信する。このステップS324の処理においては、証明書管理装置10のCPU11が第2の更新要求手段として機能する。

サーバ装置30は、この要求を受け取ると、ステップS325及びS326で、図5のステップS112及びS113の場合と同様に、従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認し、これが確認できると配布用ルート鍵証明書を証明書記憶部31に記憶する。このとき、まだ従前のルート鍵証明書は消去しない。

【0124】

そしてさらにステップS327で、図9のステップS153の場合と同様に、記憶した配布用ルート鍵証明書を用いて新ルート鍵証明書の正当性を確認する。そして、これが確認できると、次のステップS328で新ルート鍵証明書を証明書記憶部31に記憶すると共に、配布用ルート鍵証明書と従前のルート鍵証明書を廃棄する。この時点では既にクライアント装置40に新クライアント公開鍵証明書を記憶させてあるので、従前のルート鍵は不要であり、改めて廃棄要求を行うよりもここで廃棄してしまった方が処理の手順が簡単になるので、このようにしたものである。もちろん、改めて廃棄要求を行うようにしてもよい。

これらのステップS324乃至S328において、サーバ装置30のCPUが第2のサーバ側更新手段として機能する。

【0125】

次に、ステップS329及びS330で、図8のステップS14.3及びS14.4の場合と同様に、新サーバ公開鍵証明書の正当性を確認し、これが確認できると、新サーバ公開鍵証明書を証明書記憶部31に記憶し、従前のサーバ公開鍵証明書と置き換える。ただし、ここでは既に新ルート鍵証明書を記憶しているので、新クライアント公開鍵証明書の正当性は、配布用ルート鍵証明書ではなく新ル

ート鍵証明書を用いて行うことができる。これらのステップS 3 2 9及びS 3 3 0において、サーバ装置3 0のCPUが第1のサーバ側更新手段として機能する。

【0 1 2 6】

このとき従前のサーバ公開鍵証明書を消去する理由は、第1の実施形態において図7の説明で述べた通りである。そして、ステップS 3 3 0の時点では既にクライアント装置に新ルート鍵を記憶させてあるので、新サーバ公開鍵証明書を記憶させておけば、認証処理には全く問題ない。

なお、ステップS 3 2 9及びS 3 3 0の処理を、ステップS 3 2 7及びS 3 2 8の処理より前に行うようにしてもよい。この場合には、ステップS 3 2 9における正当性の確認は、配布用ルート鍵証明書を用いて行うことになる。

【0 1 2 7】

サーバ装置3 0はその後、ステップS 3 3 1で証明書管理装置1 0に対して更新要求に対する応答として結果通知を返す。

以上の図2 2に示す処理により、サーバ装置3 0側ではルート鍵更新処理が完了する。

【0 1 2 8】

その後、続いて図2 3のシーケンス図に示す処理2 3を行う。

ここではまずステップS 3 3 2で、証明書管理装置1 0がサーバ装置3 0に対して、不要になったデジタル証明書の廃棄を求める旧鍵廃棄要求をクライアント装置4 0に送信するよう要求する旧鍵廃棄要求送信要求を送信する。サーバ装置3 0は、これに応じて、クライアント装置4 0からのポーリング(S 3 3 3)に対する応答として旧鍵廃棄要求を送信するようにしている(S 3 3 4)。

以上の処理により、証明書管理装置1 0からクライアント装置4 0にサーバ装置3 0を介して上記の旧鍵廃棄要求が送信されることになる。

【0 1 2 9】

クライアント装置4 0は、この要求を受け取ると、ステップS 3 3 5で、証明書記憶部4 1に記憶している配布用ルート鍵証明書、従前のルート鍵証明書、および従前のクライアント公開鍵証明書を廃棄する。この時点では、サーバ装置3

0 に新ルート鍵証明書及び新サーバ公開鍵証明書が記憶されているので、これらの証明書を消去しても相互認証に影響はない。

クライアント装置 4 0 はその後、ステップ S 3 3 6 で証明書管理装置 1 0 に対して更新要求に対する応答として結果通知を返すが、これはまずサーバ装置 3 0 に対して送信し、サーバ装置 3 0 がステップ S 3 3 7 で証明書管理装置 1 0 に対して送信する。

以上により、ルート鍵更新処理を終了する。

【0 1 3 0】

このデジタル証明書管理システムにおいても、ルート鍵更新処理をこのような手順で行うことにより、第 1 の実施形態の場合と同様に、サーバ装置 3 0 とクライアント装置 4 0 との間の相互認証処理に大きな影響を与えることなく、ルート鍵を自動制御で更新することができる。従って、このようなデジタル証明書管理システムを用いることにより、クライアント・サーバシステムにおける S S L による相互認証を、低コストで運用することができる。

【0 1 3 1】

なお、この実施形態では、サーバ装置 3 0 に新ルート鍵を記憶させる前にクライアント装置 4 0 に新クライアント公開鍵証明書を記憶させるので、サーバ装置 3 0 に新ルート鍵を記憶させるまでは、通信に、新クライアント公開鍵証明書のデジタル署名をサーバ装置 3 0 が復号化できないことによるオーバーヘッドが生じる。しかし一方で、証明書管理装置 1 0 からサーバ装置 3 0 （あるいはサーバ装置 3 0 を介してクライアント装置 4 0 ）に計 3 回の要求を送信するのみでルート鍵の更新処理を行うことができる。従って、6 回の要求送信が必要な第 1 の実施形態の場合と比較して、処理手順の管理やプログラムの設計が容易であるという効果がある。ルート鍵証明書を更新すべきサーバ装置やクライアント装置の数が多い場合には、この効果はより大きくなり、この実施形態が有効である。

また、処理 2 1 や処理 2 2 において、各証明書について正当性を確認した後で必要なものを一括して記憶するようにすれば、証明書を記憶する不揮発メモリへのアクセス回数を低減し、処理負荷を低減すると共に処理を高速化することができる。

【0 1 3 2】

〔第 4 の実施形態：図 2 0，図 2 4 乃至図 2 6〕

次に、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント装置及びサーバ装置とによって構成される、この発明のデジタル証明書管理システムの第 4 の実施形態の構成について説明する。

このデジタル証明書管理システムは、ルート鍵更新処理の内容が第 2 の実施形態のデジタル証明書管理システムと異なるのみであり、装置の構成は第 2 の実施形態のものと同様であるのでその説明は省略する。

【0 1 3 3】

このデジタル証明書管理システムにおけるルート鍵更新動作は、この発明のデジタル証明書管理方法の第 4 の実施形態に係る動作であり、図 2 0 及び図 2 4 乃至図 2 6 のシーケンス図に示す処理 T 及び処理 3 1 乃至 3 3 を、この順で実行するものである。そしてこれらの処理は、証明書管理装置 1 0，サーバ装置 3 0，クライアント装置 4 0 の各 CPU が、所要の制御プログラムを実行することによって行うものである。

【0 1 3 4】

また、これらの処理は、図 2 0 に示す部分については第 3 の実施形態の場合と共通であり、図 2 4 乃至図 2 6 に示す部分については、第 3 の実施形態で図 2 1 乃至図 2 3 を用いてそれぞれ説明した処理と同じ目的の処理であり、証明書管理装置 1 0 と直接通信する装置がクライアント装置 4 0 であることに伴って、第 2 の実施形態で図 1 3 及び図 1 4 を用いて説明した処理 1 1 及び処理 1 2 の場合と同様に通信手順を若干変更したのみである。そこで、これらの処理についての詳細な説明は省略する。

【0 1 3 5】

そして、この第 4 の実施形態のデジタル証明書管理システムにおいても、ルート鍵更新処理をこのような手順で行うことにより、証明書管理装置 1 0 がクライアント・サーバシステムを構成する装置のうちクライアント装置 4 0 のみと通信可能な場合でも、第 3 の実施形態の場合と同様に、サーバ装置 3 0 とクライアン

ト装置 40 との間の相互認証処理に大きな影響を与えることなくルート鍵を自動制御で更新することができる。従って、このようなデジタル証明書管理システムを用いることにより、クライアント・サーバシステムにおける SSL による相互認証を、低コストで運用することができる。また、処理手順の管理やプログラムの設計が容易であるという効果もある。

【0136】

〔第5の実施形態：図27乃至図31〕

次に、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント装置及びサーバ装置とによって構成される、この発明のデジタル証明書管理システムの第5の実施形態の構成について説明する。

このデジタル証明書管理システムにおいては、図27に示すように、クライアント・サーバシステムを1つのサーバ装置と複数のクライアント装置とによって構成している。個々の証明書管理装置10、サーバ装置30、クライアント装置40の構成は第1の実施形態の場合と同様であるので、詳細な図示及び説明は省略するが、複数のクライアント装置40-1～nを、サーバ装置30と通信可能なように設けている。そして、証明書管理装置10と各クライアント装置40との通信は、サーバ装置30が仲介して行う。

【0137】

ところで、このデジタル証明書管理システムにおいては、証明書管理装置10の構成記憶部26に、クライアント・サーバシステムを構成する各ノードの情報を、図28に示す形式で記憶している。すなわち、各ノード毎に、ノードID、デジタル証明書管理装置(CA)10との直接通信の可否、および、そのノードの通信相手となる各ノードのIDと共に、その通信相手と通信する際にクライアントとサーバのいずれとして機能するかを示す情報、その通信相手と通信する際に使用するルート鍵の情報、その通信相手におけるルート鍵証明書及び公開鍵証明書の更新状態を示す情報を記憶している。ここで、「通信相手」とは、認証を行った上で通信を行う相手を指すものとする。また、図示は省略したが、各ノードの情報として、そのノードが保有しているルート鍵証明書や公開鍵証明書のI

Dを、その有効期限と共に記憶するようにしてもよい。

【0138】

図28に示した形式で記憶する具体的な情報としては、例えば図27に示すサーバ装置30については、図29(a)に示すような情報を記憶することになる。すなわち、ノードIDとして「サーバ装置30」を記憶し、証明書管理装置10と直接通信可能であるのでその旨の情報を記憶している。そして、通信相手となるノードの情報として、クライアント装置40-1～nの情報をそれぞれ記憶している。また、これらの各装置と通信する際に、サーバ装置30はサーバとして機能するのでその旨を記憶し、使用するルート鍵の情報としてはここでは「ルート鍵A」を記憶している。そして、ルート鍵Aは更新が必要な状態であるとし、その旨の情報も記憶している。

【0139】

各クライアント装置40-1～nについては、図29(b)と(c)の双方の記録形態が考えられる。図にはクライアント装置40-1についての記憶例を示しているが、ノードIDとして「クライアント装置40-1」を記憶し、証明書管理装置10とはサーバ装置30を介して通信するので直接通信不能である旨の情報を記憶する点は、どちらも共通であるが、通信相手となるノードの情報の記録形態が異なる。

【0140】

すなわち、(b)の形態では、サーバ装置30とクライアント装置40-1とが通信可能であることは、サーバ装置30に関する情報として(a)に記録済みであり、サーバ／クライアントの別等もその情報から導き出せるので、クライアント装置40-1に関する情報として新たに記憶することはしていない。一方(c)の形態では、クライアント装置40-1に関する情報として別途サーバ装置30とクライアント装置40-1とが通信可能であることを記憶するようにしている。

(b)の形態では情報の記憶容量が少なく済み、(c)の形態では対象ノードについての情報のみを参照すればそのノードの通信相手を知ることができる。しかし、どちらの形態を取るにせよ、各ノードの通信相手及びその通信相手との

間でクライアントとサーバのいずれとして機能するかの情報は記憶できているので、これを参照して後述のように証明鍵の更新手順を定めることができる。

【0141】

次に、図27に示した第5の実施形態のデジタル証明書管理システムにおけるルート鍵更新処理について説明する。この処理は、この発明のデジタル証明書管理方法の第5の実施形態に係る処理である。

この処理は、基本的には、第1の実施形態で説明した処理S及び処理1乃至6を図31のフローチャートに示す順番で実行するものである。そしてこれらの処理は、証明書管理装置10，サーバ装置30，クライアント装置40-1～nの各CPUが、所要の制御プログラムを実行することによって行うものである。

ただし、この実施形態においては、クライアント装置40を複数設けているので、これに伴ってクライアント装置40に対して行う処理が若干異なったものになる。すなわち、各クライアント装置40毎に、個別に配布用ルート鍵証明書や新クライアント証明書、新ルート鍵証明書を送信して記憶させるようにする必要があるのである。

【0142】

図30に、図7に示したクライアント装置の公開鍵証明書記憶処理をクライアント装置40-1に対して行う場合の処理シーケンスを処理3-1として示す。この図からわかるように、処理の流れ自体は図7に示した処理と変わらない。図30に示した各処理は、図7に示した処理のうちステップ番号の下2ケタが一致する処理と対応するものである。しかし、ステップS531で作成する新クライアント公開鍵証明書は、クライアント装置40-1が用いるためのものであり、ステップS532における更新要求送信要求においても、更新要求の送信先としてクライアント装置40-1を指定している。

【0143】

このような処理は、当然他のクライアント装置40-2～nについても行うが、実行時期についての条件が満たされていれば、初めの装置についての公開鍵証明書記憶処理に対する応答が帰ってくる前に次の装置についての公開鍵証明書記憶処理を行っても問題ない。また、複数の装置についての公開鍵証明書記憶処理

をまとめ、ステップS532でそれらの各装置に対応する新クライアント公開鍵証明書と各更新要求の送信先とを1つのメッセージに含める形でサーバ装置30に送信するようにしてもよい。この場合でもステップS533乃至S537の処理をクライアント装置毎に行うことはもちろんであるが、ステップS538の結果通知については、クライアント装置毎に送信するようにしても、複数の装置からの結果通知を1つのメッセージに含める形で送信するようにしてもよい。

【0144】

なお、ここでは処理3に関する相違点について説明したが、図6に示した処理2及び図10に示した処理6についても同様な点が第1の実施形態の場合と異なる。サーバ装置30については、1つしか設けていないので、サーバ装置30に対して行う処理1, 4, 5は第1の実施形態の場合と同様である。

また、上記の処理3-1という番号は、クライアント装置40-1に対する処理3に相当する処理という意味で付したものであり、以下の説明においても、処理の番号は装置に付した符号の添え字を用いて同様に付すものとする。例えば、クライアント装置40-nに対する処理3に相当する処理は処理3-n, クライアント装置40-1に対する処理6に相当する処理は処理6-1等である。

【0145】

このデジタル証明書管理システムにおける各処理の実行タイミングは、図31に示すフローチャートのようなものになる。すなわち、ルート鍵の更新を行う場合には、まず図4に示した処理Sを実行し、その後処理1乃至処理6を実行する。

図31の記載から明らかなように、この第5の実施形態におけるルート鍵更新処理は、図11に示した第1の実施形態の場合の実行タイミングと概ね同様のものである。しかし、処理2, 3, 6を各クライアント装置に対して行う必要があることに伴い、若干異なったものになっている。

【0146】

具体的には、処理1及び処理2-1~nは処理Sの完了後に開始する。処理3-1~nは、処理2-1~nのうち対応する処理の完了後に開始する（例えば、処理3-1は処理2-1の完了後に開始する）が、処理1も完了した後に開始す

る方が好ましい。処理 4 は、処理 1 及び処理 2 - 1 ~ n の全てが完了した後に開始する。処理 5 は、処理 1 及び処理 3 - 1 ~ n の全てが完了した後に開始する。処理 6 は、処理 2 - 1 ~ n のうち対応する処理及び処理 4 の完了後に開始する。そして、処理 3 - 1 ~ n, 処理 4, 処理 5, 処理 6 - 1 ~ n が全て完了した時点で、ルート鍵及び公開鍵証明書の更新が終了したことになる。

なお、処理 2, 4, 6 については、それぞれの開始条件が満たされれば、各クライアント装置に対する処理は任意の順番で行って構わない。

【0 1 4 7】

図 3 1 に示す処理手順において、この実施形態の特徴となるのは、まず、処理 4 (サーバ装置の公開鍵証明書記憶処理) を、全てのクライアント装置 4 0 について処理 2 (クライアント装置のルート鍵証明書記憶処理) が完了した後で、すなわちサーバ装置 3 0 の通信相手となる全てのクライアント装置 4 0 から配布用ルート鍵証明書を記憶した旨の応答があった後で実行する点である。

第 1 の実施形態で説明したように、サーバ装置 3 0 については新サーバ公開鍵証明書を記憶させる際に従前のものを廃棄する必要があるので、通信相手となる全てのクライアント装置 4 0 に新ルート鍵を記憶させる前にこれを行ってしまうと、認証処理に支障が生じるためである。逆に言えば、全てのクライアント装置 4 0 に新ルート鍵を記憶させた後であれば、サーバ装置 3 0 の従前のサーバ公開鍵証明書を廃棄してしまっても、認証処理に支障が生じることがない。

【0 1 4 8】

また、処理 3 - 1 ~ n (クライアント装置の公開鍵証明書記憶処理) を、処理 1 (サーバ装置のルート鍵証明書記憶処理) の後で、すなわち各クライアント装置 4 0 - 1 ~ n について、その通信相手となる全てのサーバ装置 3 0 (ここでは 1 つだけ) から配布用ルート鍵証明書を記憶した旨の応答があった後で実行するようにするとよい。第 1 の実施形態で説明したように、クライアント装置 4 0 に新クライアント公開鍵証明書を記憶させた時点で通信相手となるサーバ装置 3 0 に新ルート鍵が記憶されていないと、そのサーバ装置 3 0 に新ルート鍵が記憶されるまで通信にオーバーヘッドが生じ、効率が悪くなってしまうためである。

【0 1 4 9】

その他の点も、クライアント装置 40 を複数設けたことに伴って若干異なるが、概ね第 1 の実施形態の場合と同様であり、ルート鍵更新処理をこのような手順で行うことにより、第 1 の実施形態の場合と同様に、サーバ装置 30 と各クライアント装置 40-1 ~ n との間の相互認証処理に大きな影響を与えることなく、ルート鍵を自動制御で更新することができる。

従って、このようなデジタル証明書管理システムを用いることにより、クライアント・サーバシステムにおける SSL による相互認証を、低コストで運用することができる。

ここで、処理 5 (サーバ装置のルート鍵証明書置き換え処理) を、処理 3-1 ~ n の後で、すなわちサーバ装置 30 の通信相手となる全てのクライアント装置 40-1 ~ n から新クライアント公開鍵証明書を記憶した旨の応答があった後で実行するようにするとよい。また、処理 6-1 ~ n (クライアント装置のルート鍵証明書置き換え処理) を、処理 4 の後で、すなわち各クライアント装置 40-1 ~ n について、その通信相手となる全てのサーバ装置 30 (ここでは 1 つだけ) から新サーバ公開鍵証明書を記憶した旨の応答があった後で実行するようにするとよい。

【0150】

なお、図 31 に示したような更新手順は、証明書管理装置 10 の更新順制御部 27 が構成記憶部 26 に記憶している情報をもとに作成して管理する。そして、この更新手順の作成は、この発明の更新手順決定方法に係る処理である。この実施形態の場合には、まず証明書管理装置 10 と直接通信可能なサーバ装置 30 に関する情報を参照すると、このサーバ装置 30 はサーバとして機能し、これと通信可能なノードとしてはクライアント装置 40-1 ~ n があることがわかる。そして、全てのノードとの通信に同じルート鍵 A を使用し、更新が必要であることもわかる。さらに、各クライアント装置 40-1 ~ n に関する情報を参照すると、このクライアント・サーバシステムにはそれ以上ノードがないことがわかるので、これらの情報から更新手順を作成することができる。

【0151】

すなわち、まずサーバ装置 30 及びクライアント装置 40-1 ~ n に配布用ル

ート鍵証明書を記憶させ、これらが全て終了したらサーバ装置 30 に新サーバ公開鍵証明書を記憶させ、・・・、といったように、図 3 1 に示した条件を満たすように更新に必要な各処理の実行順序を定めればよいのである。あるいは、処理 4 の実行には処理 1 及び処理 2-1~n の全てが完了していることが必要等、各処理について実行条件を定め、これが満たされた場合にその処理を開始するように制御しても、更新手順を定めることができる。

【0152】

〔第 5 の実施形態の変形例：図 3 2 乃至図 3 4〕

以上説明した第 5 の実施形態では、ルート鍵更新処理を図 3 1 に示す手順で行う例について説明した。この処理手順は、必要最低限の条件のみを定めたものであるが、この条件のみに従うとすると、各処理の実行順序の決定や処理の進行状況の管理に当たって管理すべき情報が多くなる。そこで、ルート鍵更新処理を図 3 2 又は図 3 3 に示す手順で行うようにしてもよい。これらの図における矢印の意味は、図 3 1 の場合と同様である。

【0153】

まず、図 3 2 に示す例では、処理 1 及び処理 2-1~n は処理 S の完了後に開始し、処理 3-1~n は、これらの全ての処理の完了後に開始する。処理 4 は、処理 3-1~n の全てが完了した後に開始する。そして、処理 5 及び処理 6-1~n は、処理 4 の完了後に開始する。そして、処理 5 及び処理 6-1~n が全て完了した時点で、ルート鍵及び公開鍵証明書の更新が終了したことになる。

このようにすれば、処理 5 や処理 6 の実行に当たって処理 1 や処理 2 の実行状況を監視する必要がない。処理 4 が完了している場合には、この処理の実行条件から、処理 1 と処理 2 も共に完了していることが保証されるためである。また、処理 4 の実行に当たっても、処理 3-1~n のみの完了を確認すればよい。従って、処理の進行状況の管理を単純化することができる。

各処理の実行順序を決定する際にも、全てのノードに配布用新ルート鍵証明書を記憶させてから、クライアント装置→サーバ装置の順で新公開鍵証明書を記憶させるように定めればよいので、処理を単純化し、装置やプログラムの開発コストを低減することができる。

【0154】

また、図33に示す例では、処理S、処理1、処理2-1～n、処理3-1～n、処理4をこの順で実行し、処理5及び処理6-1～nは、処理4の完了後に開始する。そして、処理5及び処理6-1～nが全て完了した時点で、ルート鍵及び公開鍵証明書の更新が終了したことになる。

このようにすれば、図32に示した例の場合よりも、さらに処理の単純化を図ることができる。

一方で、図32及び図33に示した手順であっても、図31に示した実行順序の条件は破線で示したものも含めて全て満たしているので、クライアント・サーバシステムにおける相互認証機能の維持という観点では、上述した第5の実施形態と同等の効果を有する。

【0155】

なお、図33に示す手順に従う場合、処理2と処理3について、同一のクライアント装置40に対して行う処理をまとめて行うことができる。この場合の処理例を図34のシーケンス図に示す。ここでは、クライアント装置40-1に対してルート鍵証明書記憶処理と公開鍵証明書記憶処理とをまとめて行う場合の処理を処理2'-1として示している。

この処理においては、証明書管理装置10がステップS621で、図30のステップS531の場合と同様にクライアント40-1のための新クライアント証明書を作成する。そして、ステップS622で証明書管理装置10がサーバ装置30に対して、図4に示す処理SのステップS102で作成した配布用ルート鍵証明書と、ステップS621で作成した新クライアント公開鍵証明書と共に、これらについての更新要求をクライアント装置40-1に送信するよう要求する更新要求送信要求を送信する。

【0156】

サーバ装置30はこれに応じて、図6のステップS122及びS123の場合と同様に、クライアント装置40-1からのポーリング(S623)に対する応答としてこれらの証明書とそれらについての更新要求とを送信するようにしている(S624)。

これらの処理により、証明書管理装置 10 からクライアント装置 40-1 にサーバ装置 30 を介して上記の各証明書とそれらについての更新要求とが送信されることになり、ステップ S 6 2 2 の処理においては、証明書管理装置 10 の CPU 11 が第 1 の更新要求手段として機能する。

クライアント装置 40 は、この要求を受け取ると、ステップ S 6 2 5 及び S 6 2 6 で、図 6 のステップ S 1 2 4 及び S 1 2 5 の場合と同様に、従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認し、これが確認できると配布用ルート鍵証明書を証明書記憶部 41 に記憶する。このとき、まだ従前のルート鍵証明書は消去しない。これらの処理において、クライアント装置 40-1 の CPU が第 2 のクライアント側更新手段として機能する。

【0157】

次に、ステップ S 6 2 7 及び S 6 2 8 で、図 7 のステップ S 1 3 5 及び S 1 3 6 の場合と同様に、配布用ルート鍵証明書を用いて新クライアント公開鍵証明書の正当性を確認し、これが確認できると、新クライアント公開鍵証明書を証明書記憶部 41 に記憶する。このとき、まだ従前のクライアント公開鍵証明書は消去しない。これらの処理において、クライアント装置 40-1 の CPU が第 1 のクライアント側更新手段として機能する。

クライアント装置 40-1 はその後、ステップ S 6 2 9 で、証明書管理装置 10 に対して更新要求に対する応答として結果通知を返すが、これはまずサーバ装置 30 に対して送信し、サーバ装置 30 がステップ S 6 3 0 で証明書管理装置 10 に対して送信する。

【0158】

以上のように各クライアント装置 40 について処理 2 と処理 3 をまとめて処理 2' を行うことにより、第 3 の実施形態の場合のように、処理手順の管理やプログラムの設計が容易であるという効果がある。ここでは、クライアント装置側の処理についてはまとめていないので、この効果は後述する第 7 の実施形態の場合よりは小さいが、サーバ装置に新ルート鍵を記憶させてからクライアント装置 40 に新クライアント公開鍵証明書を記憶させているので、通信のオーバーヘッドは生じないようにすることができる。

【0159】

〔第6の実施形態：図35乃至図38〕

次に、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント装置及びサーバ装置とによって構成される、この発明のデジタル証明書管理システムの第6の実施形態の構成について説明する。

このデジタル証明書管理システムにおいては、図35に示すように、クライアント・サーバシステムを1つのクライアント装置と複数のサーバ装置とによって構成している。個々の証明書管理装置10、サーバ装置30、クライアント装置40の構成は第2の実施形態の場合と同様であるので、詳細な図示及び説明は省略するが、複数のサーバ装置30-1～nを、クライアント装置40の通信相手となるように設けている。そして、証明書管理装置10と各サーバ装置30との通信は、クライアント装置40が仲介して行う。

【0160】

このようにクライアント・サーバシステムを構成した場合には、証明書管理装置10の構成記憶部26に記憶する、クライアント・サーバシステムを構成する各ノードの情報は、図36に示すようなものになる。

すなわち、まずクライアント装置40について、図36(a)に示すような情報を記憶することになる。ここでは、ノードIDとして「クライアント装置40」を記憶し、証明書管理装置10と直接通信可能であるのでその旨の情報を記憶している。そして、通信相手となるノードの情報として、サーバ装置30-1～nの情報をそれぞれ記憶している。また、これらの各装置と通信する際に、クライアント装置40はクライアントとして機能するのでその旨を記憶し、使用するルート鍵の情報としてはここでは「ルート鍵A」を記憶している。そして、ルート鍵Aは更新が必要な状態であるとし、その旨の情報も記憶している。

【0161】

各サーバ装置30-1～nについて、図36の(b)と(c)の双方の記録形態が考えられることは第5の実施形態の場合と同様である。図にはサーバ装置30-1についての記憶例を示しているが、ノードIDとして「サーバ装置30-

1」を記憶し、証明書管理装置 10 とはクライアント装置 40 を介して通信するので直接通信不能である旨の情報を記憶している。

そして、これらの情報を参照して証明書管理装置 10 の更新順制御部 27 が証明鍵の更新手順を定めることができる。

なお、クライアント装置 40 の場合には、通信相手のサーバ装置毎に認証処理に用いるルート鍵が異なる場合が考えられるが、この場合には、共通のルート鍵を使用するグループ毎に更新処理を行うものとする。すなわち、後述するものも含め、第 1 乃至第 8 の実施形態及びそれらの変形例をグループ毎に適用することにより、グループ毎に独立して更新処理を行うことができる。

【0162】

次に、図 35 に示した第 6 の実施形態のデジタル証明書管理システムにおけるルート鍵更新処理について説明する。この処理は、この発明のデジタル証明書管理方法の第 6 の実施形態に係る処理である。

この処理は、基本的には、第 2 の実施形態で説明した処理 S 及び処理 11 乃至 16 を図 38 のフローチャートに示す順番で実行するものである。そしてこれらの処理は、証明書管理装置 10、サーバ装置 30-1~n、クライアント装置 40 の各 CPU が、所要の制御プログラムを実行することによって行うものである。

ただし、この実施形態においては、サーバ装置 30 を複数設けているので、これに伴ってサーバ装置 30 に対して行う処理が若干異なったものになる。すなわち、各サーバ装置 30 毎に、個別に配布用ルート鍵証明書や新クライアント証明書、新ルート鍵証明書を送信して記憶させるようにする必要があるのである。

【0163】

図 37 に、図 16 に示したサーバ装置の公開鍵証明書記憶処理をサーバ装置 30-1 に対して行う場合の処理シーケンスを処理 14-1 として示す。この図からわかるように、処理の流れ自体は図 16 に示した処理と変わらない。図 37 に示した各処理は、図 16 に示した処理のうちステップ番号の下 2 ケタが一致する処理と対応するものである。しかし、ステップ S741 で作成する新クライアント公開鍵証明書は、サーバ装置 30-1 が用いるためのものであり、ステップ S

742における更新要求送信要求においても、更新要求の送信先としてサーバ装置30-1を指定している。

このように、図15に示した処理14と図37に示した処理14-1との対応関係は、第5の実施形態で説明した処理3と処理3-1との対応関係と同様のものであり、処理14と比較した場合のその他の相違点も、第5の実施形態で処理3-1について説明したものと同様である。

【0164】

なお、図13に示した処理11及び図17に示した処理15についても同様な点が第2の実施形態の場合と異なる。クライアント装置40は1つしか設けていないので、クライアント装置40に対して行う処理12, 13, 16は第2の実施形態の場合と同様である。

また、上記の処理14-1という番号は、サーバ装置30-1に対する処理14に相当する処理という意味で付したものであり、以下の説明においても、処理の番号は装置に付した符号の添え字を用いて同様に付すものとする。

【0165】

このデジタル証明書管理システムにおける各処理の実行タイミングは、図38に示すフローチャートのようなものになる。すなわち、ルート鍵の更新を行う場合には、まず図4に示した処理Sを実行し、その後処理11乃至処理16を実行する。

図38の記載から明らかなように、この第6の実施形態におけるルート鍵更新処理は、図19に示した第2の実施形態の場合の実行タイミングと概ね同様のものである。しかし、処理11, 14, 15を各サーバ装置に対して行う必要があることに伴い、若干異なったものになっている。

【0166】

具体的には、処理11-1～n及び処理12は処理Sの完了後に開始する。処理13は、処理12の完了後に開始するが、処理11-1～nが全て完了した後に開始する方が好ましい。処理14-1～nは、処理12及び処理11-1～nのうち対応する処理の完了後に開始する（例えば、処理14-1は処理11-1の完了後に開始する）。処理15は、処理11-1～nのうち対応する処理及び

処理 1 3 の完了後に開始する。処理 1 6 は、処理 1 2 及び処理 1 4 - 1 ~ n が全て完了した後に開始する。そして、処理 1 3, 処理 1 4 - 1 ~ n, 処理 1 5 - 1 ~ n, 処理 1 6 が全て完了した時点で、ルート鍵及び公開鍵証明書の更新が終了したことになる。

この処理は、証明書管理装置 1 0 と直接通信する装置がクライアント装置 4 0 であることと、クライアント装置 4 0 でなくサーバ装置 3 0 を複数設けたこととに依じて若干の相違があるが、基本的には、図 3 1 に示した第 5 の実施形態の場合と対応する処理を、同様な順序で行うものである。そして、このことによる効果も、第 5 の実施形態の場合と同様である。

【0 1 6 7】

すなわち、この第 6 の実施形態のデジタル証明書管理システムにおいては、ルート鍵更新処理をこのような手順で行うことにより、証明書管理装置 1 0 がクライアント・サーバシステムを構成する装置のうちクライアント装置 4 0 のみと直接通信可能であり、サーバ装置を複数設けた場合でも、第 5 の実施形態の場合と同様に、サーバ装置 3 0 とクライアント装置 4 0 との間の相互認証処理に大きな影響を与えることなくルート鍵を自動制御で更新することができる。従って、このようなデジタル証明書管理システムを用いることにより、クライアント・サーバシステムにおける SSL による相互認証を、低コストで運用することができる。

【0 1 6 8】

また、ルート鍵更新処理の手順にポーリング待ちを必要とする箇所がないため、処理を速やかに進め、短期間で完了させることができることは、第 2 の実施形態の場合と同様である。

その他、第 5 の実施形態の場合と同様な変形を、この第 6 の実施形態に適用することも可能である。

【0 1 6 9】

〔第 7 の実施形態：図 3 9〕

次に、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント装置及びサーバ装置とによっ

て構成される、この発明のデジタル証明書管理システムの第7の実施形態の構成について説明する。

このデジタル証明書管理システムは、ルート鍵更新処理の内容が第5の実施形態のデジタル証明書管理システムと異なるのみであり、装置の構成は第5の実施形態のものと同様であるのでその説明は省略する。

【0170】

このデジタル証明書管理システムにおけるルート鍵更新動作は、この発明のデジタル証明書管理方法の第7の実施形態に係る動作である。そしてこの処理は、基本的には、第3の実施形態で説明した図20に示す処理T及び図21乃至図23にそれぞれ示す処理21乃至23をこの順で実行するものである。そしてこれらの処理は、証明書管理装置10、サーバ装置30、クライアント装置40-1～nの各CPUが、所要の制御プログラムを実行することによって行うものである。

ただし、この実施形態においては、クライアント装置40を複数設けているので、これに伴ってクライアント装置40に対して行う処理が若干異なったものになる。すなわち、第5の実施形態の場合と同様に、各クライアント装置40毎に個別に配布用ルート鍵証明書や新クライアント証明書、新ルート鍵証明書を送信して記憶させるようにする必要があるのである。

【0171】

具体的には、図21に示した処理21と、図23に示した処理23とを、各クライアント装置毎に行う。これに伴う処理の変更内容及び処理の呼称は、第5の実施形態において説明した処理3と処理3-1との対応関係と同様であるので、図示は省略するが、例えば処理21-1において図21に示す処理21のステップS311に相当する処理で作成する新クライアント公開鍵証明書は、クライアント装置40-1が用いるためのものであり、ステップS312に相当する処理における更新要求送信要求においても、更新要求の送信先としてクライアント装置40-1を指定する。

【0172】

そして、ルート鍵更新処理においては、各処理は図39のフローチャートに示

すタイミングで行う。

すなわち、まず処理 T を開始し、その完了後に処理 2 1 - 1 ~ n を任意の順番で開始する。これらの全てが完了した後で処理 2 2 を開始し、その完了後に処理 2 3 - 1 ~ n を任意の順番で開始する。そして、これらの全てが完了した時点で、ルート鍵及び公開鍵証明書の更新が終了したことになる。

【0 1 7 3】

このような更新手順は、証明書管理装置 1 0 の更新順制御部 2 7 が構成記憶部 2 6 に記憶している情報をもとに作成して管理する。そして、この更新手順の作成は、この発明の更新手順決定方法に係る処理である。この実施形態の場合には、各ノードに関する情報を参照すると、クライアント・サーバシステムにおいてクライアントとして機能するノードがクライアント装置 4 0 - 1 ~ n であることがわかるので、まずこれらについて更新処理を行い、その完了後に、サーバとして機能するサーバ装置 3 0 についての更新処理を行うように更新手順を定めればよい。そして、サーバ側の更新処理も終了した後で、クライアント側の旧鍵廃棄処理を行うようにすればよいのである。

以上のような手順で更新処理を行うことにより、第 3 の実施形態の場合と同様に一部通信のオーバーヘッドが生じるが、第 5 の実施形態の場合と比較して、処理手順の管理やプログラムの設計が容易であるという効果がある。ルート鍵証明書を更新すべきノードの数が多い場合には、この効果はより大きくなり、この実施形態が有効である。

【0 1 7 4】

〔第 8 の実施形態：図 4 0〕

次に、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント装置及びサーバ装置とによって構成される、この発明のデジタル証明書管理システムの第 8 の実施形態の構成について説明する。

このデジタル証明書管理システムは、ルート鍵更新処理の内容が第 6 の実施形態のデジタル証明書管理システムと異なるのみであり、装置の構成は第 6 の実施形態のものと同様であるのでその説明は省略する。

【0 1 7 5】

このデジタル証明書管理システムにおけるルート鍵更新動作は、この発明のデジタル証明書管理方法の第 8 の実施形態に係る動作である。そしてこの処理は、基本的には、第 4 の実施形態で説明した処理 T 及び処理 3 1 乃至 3 3 をこの順で実行するものである。そしてこれらの処理は、証明書管理装置 1 0，サーバ装置 3 0 - 1 ~ n，クライアント装置 4 0 の各 CPU が、所要の制御プログラムを実行することによって行うものである。

ただし、この実施形態においては、サーバ装置 3 0 を複数設けているので、これに伴ってサーバ装置 3 0 に対して行う処理が若干異なったものになる。すなわち、第 6 の実施形態の場合と同様に、各サーバ装置 3 0 毎に個別に配布用ルート鍵証明書や新クライアント証明書、新ルート鍵証明書を送信して記憶させるようにする必要があるのである。

【0 1 7 6】

具体的には、図 2 5 に示した処理 3 2 を、各サーバ装置毎に行う。これに伴う処理の変更内容及び処理の呼称は、第 6 の実施形態において説明した処理 1 4 と処理 1 4 - 1 との対応関係と同様であるので、図示は省略するが、例えば処理 3 2 - 1 において図 2 5 に示す処理 3 2 のステップ S 4 2 0 に相当する処理で作成する新サーバ公開鍵証明書は、サーバ装置 3 0 - 1 が用いるためのものであり、ステップ S 4 2 1 に相当する処理における更新要求送信要求においても、更新要求の送信先としてサーバ装置 3 0 - 1 を指定する。

そして、ルート鍵更新処理においては、各処理は図 4 0 のフローチャートに示すタイミングで行う。

すなわち、まず処理 T を開始し、その完了後に処理 3 1 を開始する。そして、この処理が完了した後で処理 2 2 - 1 ~ n を任意の順番で開始し、その全てが完了した後に処理 2 3 を開始する。この処理が完了した時点で、ルート鍵及び公開鍵証明書の更新が終了したことになる。

【0 1 7 7】

このような更新手順は、証明書管理装置 1 0 の更新順制御部 2 7 が構成記憶部 2 6 に記憶している情報をもとに作成して管理する。そして、この更新手順の作

成は、この発明の更新手順決定方法に係る処理である。この実施形態の場合には、各ノードに関する情報を参照すると、クライアント・サーバシステムにおいてクライアントとして機能するノードがクライアント装置 4 0 であることがわかるので、まずこれについて更新処理を行い、その完了後に、このクライアント装置 4 0 の通信相手であるサーバとして機能するサーバ装置 3 0 - 1 ~ n についての更新処理を行うように更新手順を定めればよい。そして、サーバ側の更新処理も全て終了した後で、クライアント側の旧鍵廃棄処理を行うようにすればよいのである。

以上のような手順で更新処理を行うことにより、第 4 の実施形態の場合と同様に一部通信のオーバーヘッドが生じるが、第 6 の実施形態の場合と比較して、処理手順の管理やプログラムの設計が容易であるという効果がある。ルート鍵証明書を更新すべきノードの数が多い場合には、この効果はより大きくなり、この実施形態が有効である。

【0 1 7 8】

〔第 5 乃至第 8 の実施形態の変形例：図 4 1 乃至図 4 4〕

上述した第 5 乃至第 8 の実施形態では、証明書管理装置 1 0 と直接通信するノード 1 つと、そのノードの通信相手となる複数のノードとによってクライアント・サーバシステムを構成した場合の例について説明した。しかし、この発明は、図 4 1 及び図 4 2 に示すように、クライアント・サーバシステムを構成するサーバとクライアントとをそれぞれ複数設け、これらのノードのうち、複数のノードを証明書管理装置 1 0 と直接通信可能とする場合にも適用できる。

ここで、このような場合のルート鍵の更新手順について説明する。なお、図 4 1 あるいは図 4 2 に示したクライアント・サーバシステムにおいて、各ノード間の相互認証には全て同じルート鍵を使用するものとする。

【0 1 7 9】

まず、図 4 1 には、証明書管理装置 1 0 と直接通信可能なサーバ装置 3 0 を複数設けているが、全てのクライアント装置 4 0 が 1 つのみのサーバ装置 3 0 を通信相手とする場合の例を示している。このような場合には、サーバ装置毎に別々のクライアント・サーバシステムがあるものとして更新処理を行うことができる

。

すなわち、図 4 1 に示した例では、サーバ装置 3 0 - 1 及びクライアント装置 4 0 - 1 ~ 3 で構成されたクライアント・サーバシステムと、サーバ装置 3 0 - 2 及びクライアント装置 4 0 - 4 ~ 5 で構成されたクライアント・サーバシステムとに対して独立にルート鍵更新処理を行うようにすればよい。システムをまたいだ認証処理は行われないのであるから、このようにしても、各クライアント・サーバシステムに対して第 5 あるいは第 7 の実施形態で説明した更新手順で更新処理を行うようにすれば、各ノードの間での認証処理に大きな支障を来すことなく、ルート鍵の更新を行うことができる。

【0180】

また、図 4 2 には、複数のサーバ装置を通信相手とするクライアント装置（クライアント装置 4 0 - 3）が存在する場合の例を示している。このような場合には、全てのノードによって 1 つのクライアント・サーバシステムが構成されるものとして更新処理を行う必要がある。しかし、このような場合であっても、それぞれのサーバ装置に新サーバ証明書を記憶させる処理を、そのサーバ装置の通信相手となる全てのクライアント装置に対して新ルート鍵を記憶させた後で行うようにすればよいことは、第 5 及び第 7 の実施形態の場合と同様である。

【0181】

この例の場合の更新処理に必要な各処理の開始条件を図示すると、図 4 3 のようになる。この図において、各矢印や処理番号の意味は、第 5 の実施形態の説明で用いた図 3 1 の場合と同様である。クライアント・サーバシステムの構成が複雑になったことに伴い、開始条件の内容も図 3 1 と比較してかなり複雑になる。しかし、例えばサーバ装置 3 0 - 1 についての公開鍵証明書記憶処理である処理 4 - 1 は、そのサーバ装置 3 0 - 1 自身及びその通信相手となるクライアント装置 4 0 - 1 ~ 3 についてのルート鍵証明書記憶処理である処理 1 - 1 及び処理 2 - 1 ~ 3 が全て完了してから開始する等、各処理の開始条件は、図 3 1 の場合と同様な規則に基づいている。クライアント装置 4 0 - 3 についての公開鍵証明書記憶処理である処理 3 - 3 を、そのサーバ装置 4 0 - 3 自身及びその通信相手となるサーバ装置 3 0 - 1, 2 についてのルート鍵証明書記憶処理である処理 2 -

3 及び処理 1-1, 2 が全て完了してから開始するようにするとよいことも、図 31 の場合と同様な規則から導き出すことができる。

ただし、サーバ装置 30-1 とクライアント装置 40-4 等、互いに通信相手とならないノード間については、もともと認証処理は行わないのであるから、相互の証明書の記憶状況を適切な関係に保つ必要はなく、処理順序の管理も必ずしも行う必要はない。

【0182】

また、第 7 の実施形態の場合のように、各ノードにルート鍵証明書と公開鍵証明書とを一括して記憶させる場合には、各処理の開始条件は図 44 のようになる。この図は図 39 と対応するものであり、このような処理手順も、図 39 の場合と同様に、サーバ装置に対する更新処理を、そのサーバ装置の通信相手となる全てのクライアント装置に対する更新処理が完了した後で開始するという条件に従って定めることができる。

このような図 43 及び図 44 に示したような更新手順も、第 5 あるいは第 7 の実施形態の場合と同様に、証明書管理装置 10 の更新順制御部 27 が構成記憶部 26 に記憶している情報をもとに作成して管理する。クライアント・サーバシステムの構成が図 42 に示すようなものであっても、構成記憶部 26 に記憶している各ノードに関する情報を参照することにより、各ノードの通信相手及びその機能を把握することができるので、それを基に更新手順を作成することができるのである。

【0183】

ここで、更新手順を作成する場合において、クライアント装置 40-3 のように複数のサーバ装置 30 と通信可能なノードへの要求は、どちらのサーバ装置 30 を介して行うようにしてもよい。

なお、ここで説明した変形例では、証明書管理装置 10 と直接通信可能なノードがサーバ装置である例について説明したが、これがクライアント装置であっても同様な変形を適用できることはもちろんであり、この場合には、上述のような変形を第 6 あるいは第 8 の実施形態に適用することになる。

【0184】

〔上述した各実施形態についての他の変形例：図45〕

以上説明した実施形態では、クライアント装置40とサーバ装置30とが図46を用いて説明したようなSSLによる相互認証を行う場合の例について説明した。しかし、この相互認証が必ずしもこのようなものでなくてもこの発明は効果を発揮する。

SSLを改良したTLS (Transport Layer Security) も知られているが、このプロトコルに基づく認証処理を行う場合にも当然適用可能である。

【0185】

また、上述した実施形態では、証明書管理装置10をサーバ装置30あるいはクライアント装置40とは別に設ける例について説明したが、サーバ装置30あるいはクライアント装置40と一体として設けることを妨げるものではない。この場合、証明書管理装置10の機能を実現するためのCPU、ROM、RAM等の部品を独立して設けてもよいが、ハードウェア資源としてはサーバ装置30あるいはクライアント装置40のCPU、ROM、RAM等を使用し、そのCPUに適当なソフトウェアを実行させることにより、証明書管理装置10として機能させるようにしてもよい。

【0186】

このような場合において、証明書管理装置10と、これと一体になっているサーバ装置30あるいはクライアント装置40との間の通信には、ハードウェアを証明書管理装置10として機能させるためのプロセスと、ハードウェアをサーバ装置30あるいはクライアント装置40として機能させるためのプロセスとの間のプロセス間通信を含むものとする。

さらに、上述した各実施形態では、証明書管理装置10が証明鍵やデジタル証明書を自ら作成してこれを取得する例について説明したが、図2及び図12に示した証明用鍵作成部21や証明書発行部22の機能を証明書管理装置10とは別の装置に設け、証明書管理装置10がその装置から証明鍵やデジタル証明書の供給を受けてこれらを取得するようにしてもよい。

【0187】

また、証明書管理装置10がサーバ装置30及びクライアント装置40の双方

と直接的に通信が可能な構成としても構わない。この場合、図5乃至図10等
示した通信シーケンスは、双方の装置と直接通信が可能であることに伴って異な
ったものになるが、処理の順序は上述した各実施形態の場合と同様である。この
ようにしても、上述した各実施形態の効果をすることができる。

【0188】

また、上述したように、第2及び第4の実施形態においては、証明書管理装置
10とクライアント装置40との間で通信を行う際にも、SSLによる相互認証
を行うようにすることができる。

このようにするには、図45に示すように、クライアント装置40に、サーバ
装置30との相互認証に用いるクライアント秘密鍵、クライアント公開鍵証明書
及びルート鍵証明書（実施形態において説明したもの）とは別に、もう一組の秘
密鍵、公開鍵証明書及びルート鍵証明書（「第2のクライアント秘密鍵」、「第
2のクライアント公開鍵証明書」及び「第2のルート鍵証明書」と呼ぶ）を記憶
させ、証明書管理装置10との相互認証にこれらを用いるようにすればよい。

【0189】

この場合、証明書管理装置10にも、管理装置用秘密鍵、管理装置用公開鍵証
明書及び上記の第2のルート鍵証明書を記憶させ、相互認証に用いる。そして、
第2のクライアント公開鍵証明書及び管理装置用公開鍵証明書は、第2のルート
鍵証明書に含まれる第2のルート鍵で内容が確認できるものとする。すなわち、
その第2のルート鍵と対応するルート私有鍵（第2のルート私有鍵）を用いてデ
ジタル署名を付すようにする。

このようにすれば、証明書管理装置10とクライアント装置40との間の相互
認証と、クライアント装置40とサーバ装置30との間の相互認証とを、全く独
立して行うことができる。

【0190】

第2及び第4の実施形態におけるクライアント装置40は、図12を用いて説
明したように、証明書管理装置10との通信はサーバ機能部44が、サーバ装置
30との通信はクライアント機能部43が通信機能部42を介して行う。従って
、証明書管理装置10から通信を要求される通信と、サーバ装置30に要求する

通信とは明確に区別することができるため、これらとの間で別々の鍵や証明書を
用いた相互認証を行うことができるのである。

このような場合において、証明書管理装置 10 からの要求に応じてクライアント
装置 40 とサーバ装置 30 との間の相互認証に用いるルート鍵証明書や公開鍵
証明書を更新したとしても、証明書管理装置 10 とクライアント装置 40 との間
の相互認証には全く影響がない。

【0191】

各実施形態で説明した手順によって更新処理を行えば、クライアント装置 40
とサーバ装置 30 との間の相互認証にも大きな影響を与えることなく更新処理を
行えることは上述した通りであるので、図 45 に示した構成をとることにより、
各ノード間の相互認証を維持したままルート鍵を更新できると言える。

なお、第 2 のルート鍵証明書を更新しようとする場合には、証明書管理装置 1
0 をクライアント、クライアント装置 40 をサーバとして、上述したいずれかの
実施形態の手順に従って更新処理を行えばよい。このような更新処理を行っても
、クライアント装置 40 とサーバ装置 30 との間の相互認証には全く影響がない
。

第 6 及び第 8 の実施形態のように、サーバ装置 30 を複数設けた場合であって
も、同様な対応が可能である。

【0192】

また、この発明によるプログラムは、クライアント・サーバシステムを構成す
る複数の装置とネットワークを介して直接的又は間接的に通信可能なコンピュー
タに、この発明による各機能（構成記憶手段、更新順制御手段、証明鍵更新手段
、第 1 の更新要求手段、第 2 の更新要求手段、その他の手段としての機能）を実
現させるためのプログラムであり、このようなプログラムをコンピュータに実行
させることにより、上述したような効果を得ることができる。

【0193】

このようなプログラムは、はじめからコンピュータに備える ROM あるいは H
DD 等の記憶手段に格納しておいてもよいが、記録媒体である CD-ROM ある
いはフレキシブルディスク、SRAM、EEPROM、メモリカード等の不揮発

性記録媒体（メモリ）に記録して提供することもできる。そのメモリに記録されたプログラムをコンピュータにインストールしてCPUに実行させるか、CPUにそのメモリからこのプログラムを読み出して実行させることにより、上述した各手順を実行させることができる。

さらに、ネットワークに接続され、プログラムを記録した記録媒体を備える外部機器あるいはプログラムを記憶手段に記憶した外部機器からダウンロードして実行させることも可能である。

【0194】

【発明の効果】

以上説明してきた通り、この発明のデジタル証明書管理システム、デジタル証明書管理装置、デジタル証明書管理方法によれば、クライアント・サーバシステムにおける認証処理でデジタル証明書の正当性の確認に用いる証明鍵を、その認証処理に支障を来すことなく自動的に更新することができる。そして、このことにより、公開鍵暗号を利用したデジタル証明書を用いるSSL等の方式による相互認証を、クライアント・サーバシステムにおいて低コストで実現可能とすることができる。

この発明の更新手順決定方法によれば、上記のような証明鍵の更新を行うための更新処理の適切な手順を決定できるので、適当な更新装置にこの手順に従って更新処理を行わせることにより、同様な効果を得ることができる。

また、この発明のプログラムによれば、コンピュータにデジタル証明書管理装置を制御させてこのようなデジタル証明書管理装置の特徴を実現し、同様な効果を得ることができる。

【図面の簡単な説明】

【図1】

この発明のデジタル証明書管理装置の実施形態である証明書管理装置のハードウェア構成を示すブロック図である。

【図2】

この発明のデジタル証明書管理システムの第1の実施形態を構成する各装置の、この発明の特徴となる部分の機能構成を示す機能ブロック図である。

【図 3】

図 2 に示したデジタル証明書管理システムにおけるデータ送受モデルを示す概念図である。

【図 4】

図 2 に示したデジタル証明書管理システムにおけるルート鍵更新処理のうち、ルート鍵証明書作成処理を示すシーケンス図である。

【図 5】

同じくサーバ装置のルート鍵証明書記憶処理を示すシーケンス図である。

【図 6】

同じくクライアント装置のルート鍵証明書記憶処理を示すシーケンス図である。

【図 7】

同じくクライアント装置の公開鍵証明書記憶処理を示すシーケンス図である。

【図 8】

同じくサーバ装置の公開鍵証明書記憶処理を示すシーケンス図である。

【図 9】

同じくサーバ装置のルート鍵証明書書き換え処理を示すシーケンス図である。

【図 10】

同じくクライアント装置のルート鍵証明書書き換え処理を示すシーケンス図である。

【図 11】

第 1 の実施形態のルート鍵更新処理における、図 4 乃至図 10 のシーケンス図に示した各処理の実行順を示すフローチャートである。

【図 12】

この発明のデジタル証明書管理システムの第 2 の実施形態を構成する各装置の、この発明の特徴となる部分の機能構成を示す機能ブロック図である。

【図 13】

図 12 に示したデジタル証明書管理システムにおけるルート鍵更新処理のうち、サーバ装置のルート鍵証明書記憶処理を示すシーケンス図である。

【図 1 4】

同じくクライアント装置のルート鍵証明書記憶処理を示すシーケンス図である。

【図 1 5】

同じくクライアント装置の公開鍵証明書記憶処理を示すシーケンス図である。

【図 1 6】

同じくサーバ装置の公開鍵証明書記憶処理を示すシーケンス図である。

【図 1 7】

同じくサーバ装置のルート鍵証明書書き換え処理を示すシーケンス図である。

【図 1 8】

同じくクライアント装置のルート鍵証明書書き換え処理を示すシーケンス図である。

【図 1 9】

第 2 の実施形態のルート鍵更新処理における、図 4 及び図 1 3 乃至図 1 8 のシーケンス図に示した各処理の実行順を示すフローチャートである。

【図 2 0】

この発明のデジタル証明書管理システムの第 3 の実施形態におけるルート鍵更新処理の一部を示すシーケンス図である。

【図 2 1】

図 2 0 の続きの処理を示すシーケンス図である。

【図 2 2】

図 2 1 の続きの処理を示すシーケンス図である。

【図 2 3】

図 2 2 の続きの処理を示すシーケンス図である。

【図 2 4】

この発明のデジタル証明書管理システムの第 4 の実施形態におけるルート鍵更新処理の、図 2 0 の続きの処理を示すシーケンス図である。

【図 2 5】

図 2 4 の続きの処理を示すシーケンス図である。

【図 2 6】

図 2 5 の続きの処理を示すシーケンス図である。

【図 2 7】

この発明のデジタル証明書管理システムの第 5 の実施形態を構成する各装置の
関係を示すブロック図である。

【図 2 8】

図 2 に示した構成記憶部 2 6 に記憶する各ノードの情報の記憶形式の例を示す
図である。

【図 2 9】

図 2 7 に示したサーバ装置 3 0 及びクライアント装置 4 0 - 1 について、図 2
8 に示した形式で情報を記載した場合の記載例を示す図である。

【図 3 0】

第 1 の実施形態で説明した処理を第 5 の実施形態に適用する場合の変更点につ
いて説明するための図である。

【図 3 1】

第 5 の実施形態のルート鍵更新処理における、各処理の実行順を示す図 1 1 と
対応するフローチャートである。

【図 3 2】

その変形例のルート鍵更新処理における、各処理の実行順を示すフローチャ
ートである。

【図 3 3】

その別の変形例のルート鍵更新処理における、各処理の実行順を示すフローチ
ャートである。

【図 3 4】

図 3 3 に示した処理において、クライアント装置のルート鍵証明書記憶処理と
公開鍵証明書記憶処理とをまとめて行う場合の処理を示すシーケンス図である。

【図 3 5】

この発明のデジタル証明書管理システムの第 6 の実施形態を構成する各装置の
関係を示すブロック図である。

【図 3 6】

図 3 5 に示したクライアント装置 4 0 及びサーバ装置 3 0 - 1 について、図 2 8 に示した形式で情報を記載した場合の記載例を示す図である。

【図 3 7】

第 2 の実施形態で説明した処理を第 6 の実施形態に適用する場合の変更点について説明するための図である。

【図 3 8】

第 6 の実施形態のルート鍵更新処理における、各処理の実行順を示す図 1 9 と対応するフローチャートである。

【図 3 9】

第 7 の実施形態のルート鍵更新処理における、各処理の実行順を示すフローチャートである。

【図 4 0】

第 8 の実施形態のルート鍵更新処理における、各処理の実行順を示すフローチャートである。

【図 4 1】

この発明のデジタル証明書管理システムの第 5 乃至第 8 の実施形態に適用する変形例を構成する各装置の関係を示すブロック図である。

【図 4 2】

その別の変形例を構成する各装置の関係を示すブロック図である。

【図 4 3】

図 4 2 に示した構成のデジタル証明書管理システムにおけるルート鍵更新処理を構成する各処理の開始条件を示す図である。

【図 4 4】

同じく、各ノードにルート鍵証明書と公開鍵証明書とを一括して記憶させる場合の各処理の開始条件を示す図である。

【図 4 5】

別の変形例における、鍵及び証明書の記憶状態及びその場合のルート鍵更新処理について説明するための図である。

【図 4 6】

クライアント装置とサーバ装置とがSSLによる相互認証を行う際の各装置において実行する処理のフローチャートを、その処理に用いる情報と共に示す図である。

【図 4 7】

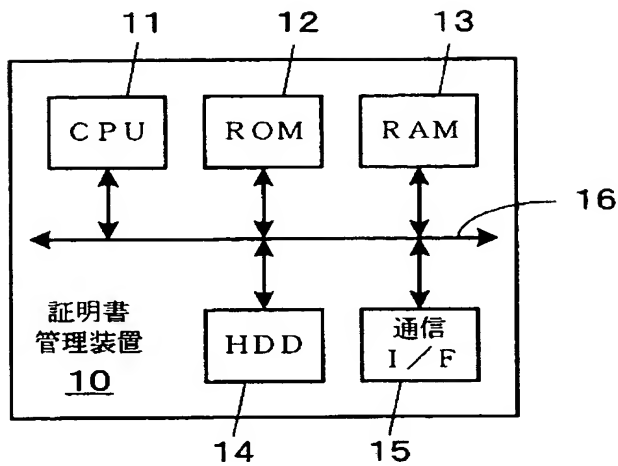
図 4 6 に示した認証処理におけるルート鍵、ルート私有鍵、およびクライアント公開鍵の関係について説明するための図である。

【符号の説明】

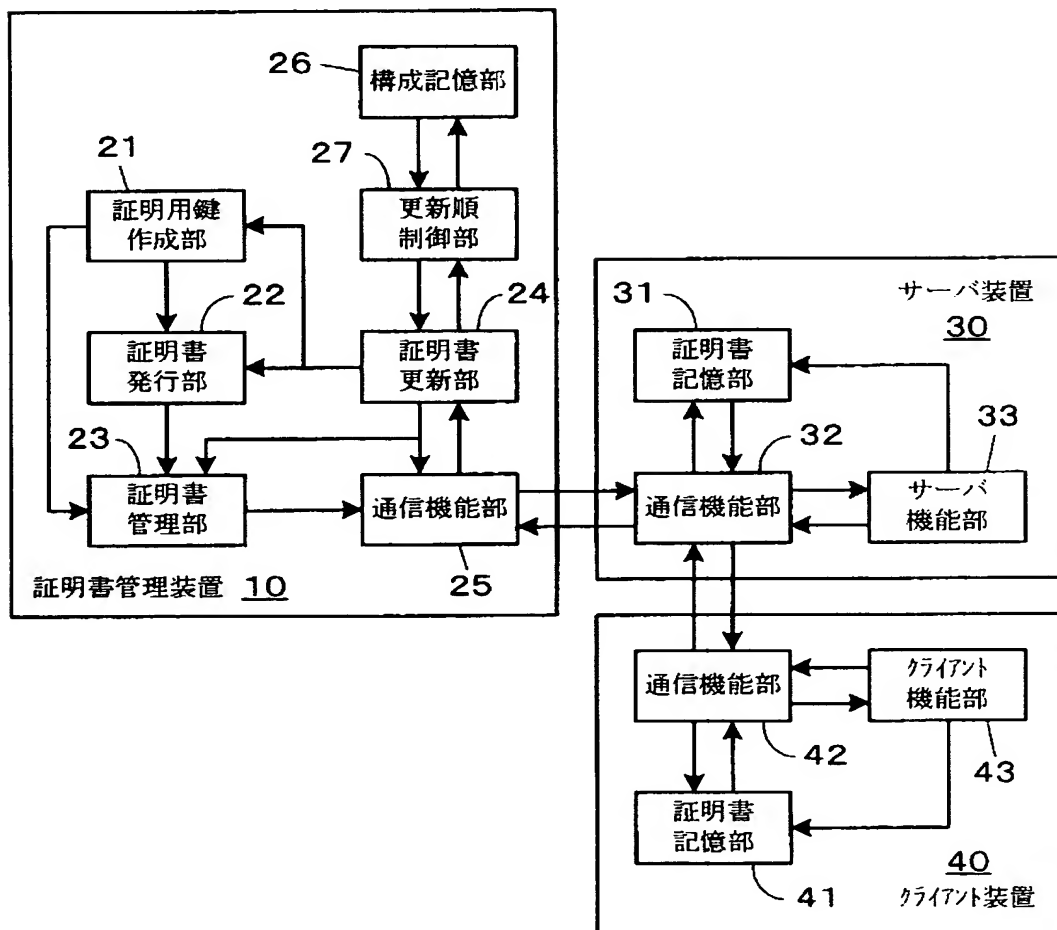
10：証明書管理装置	11：CPU
12：ROM	13：RAM
14：HDD	15：通信 I/F
16：システムバス	21：証明用鍵作成部
22：証明書発行部	23：証明書管理部
24：証明書更新部	25, 32, 42：通信機能部
30：サーバ装置	31, 41：証明書記憶部
33, 44：サーバ機能部	40：クライアント装置
43：クライアント機能部	

【書類名】 図面

【図 1】

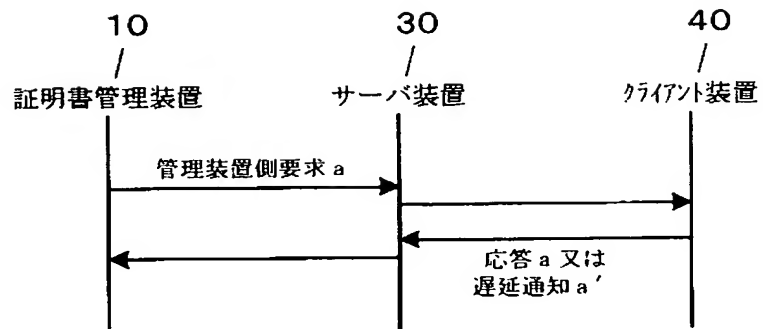


【図 2】

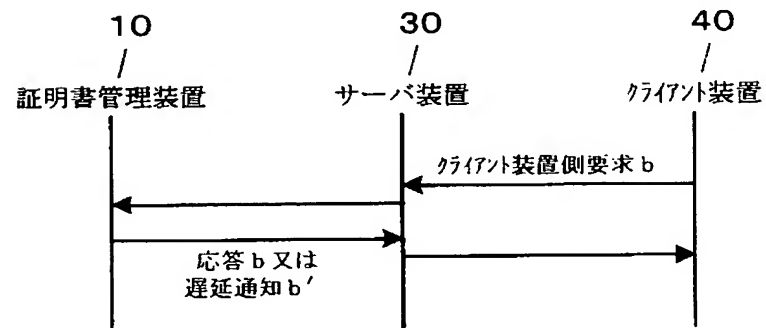


【図 3】

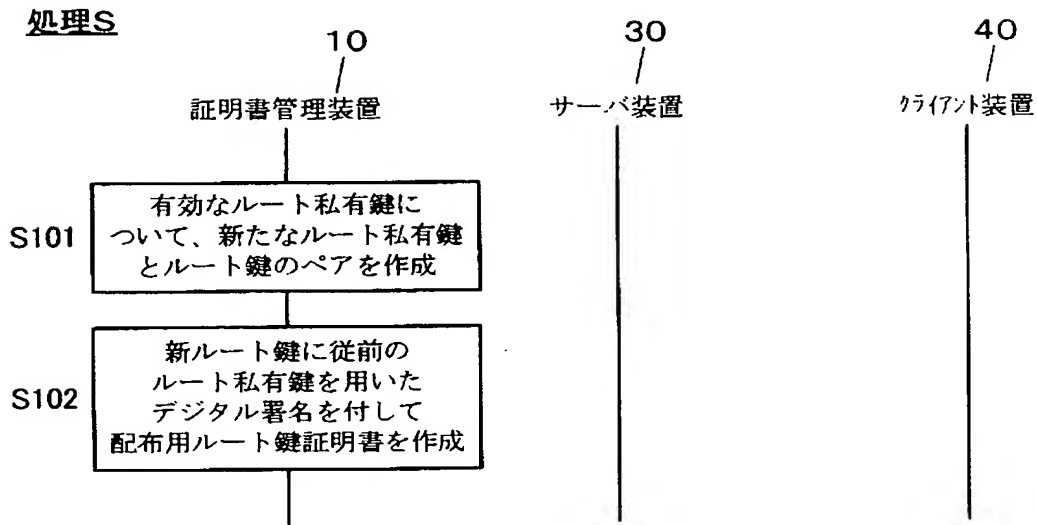
(A)



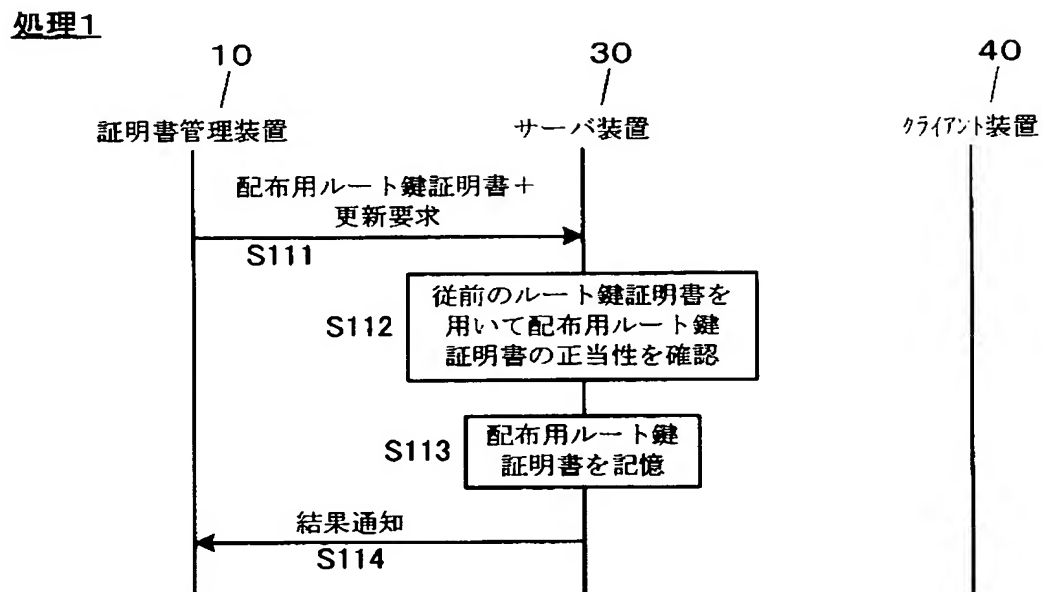
(B)



【図 4】

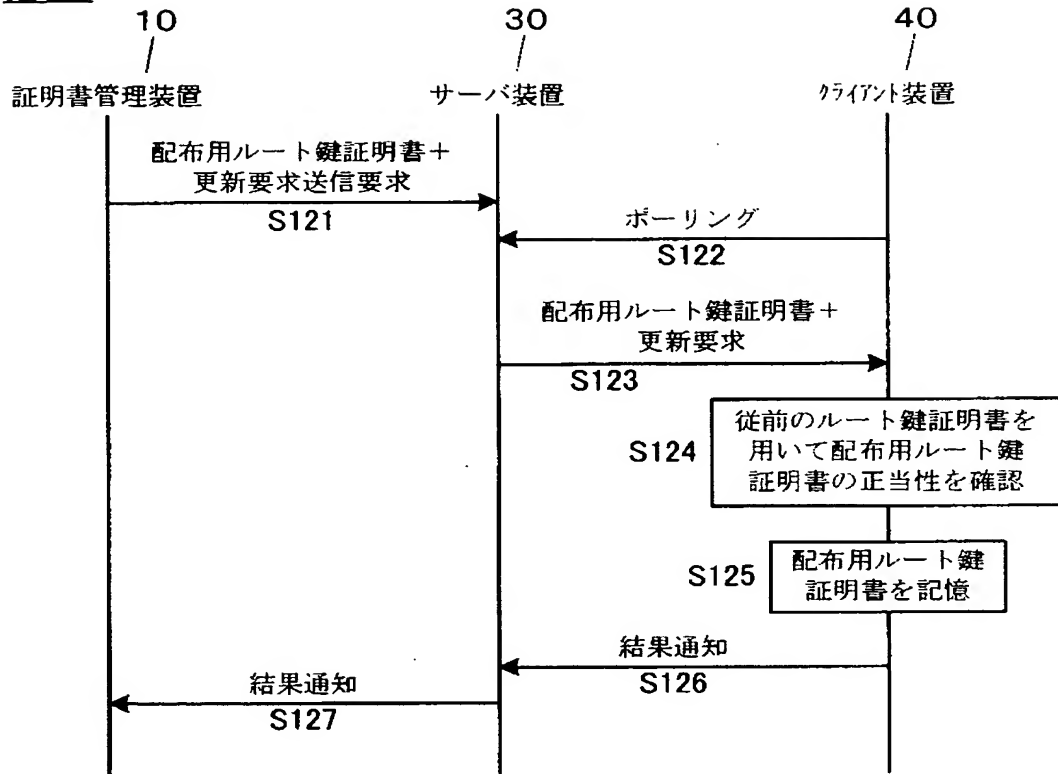


【図 5】



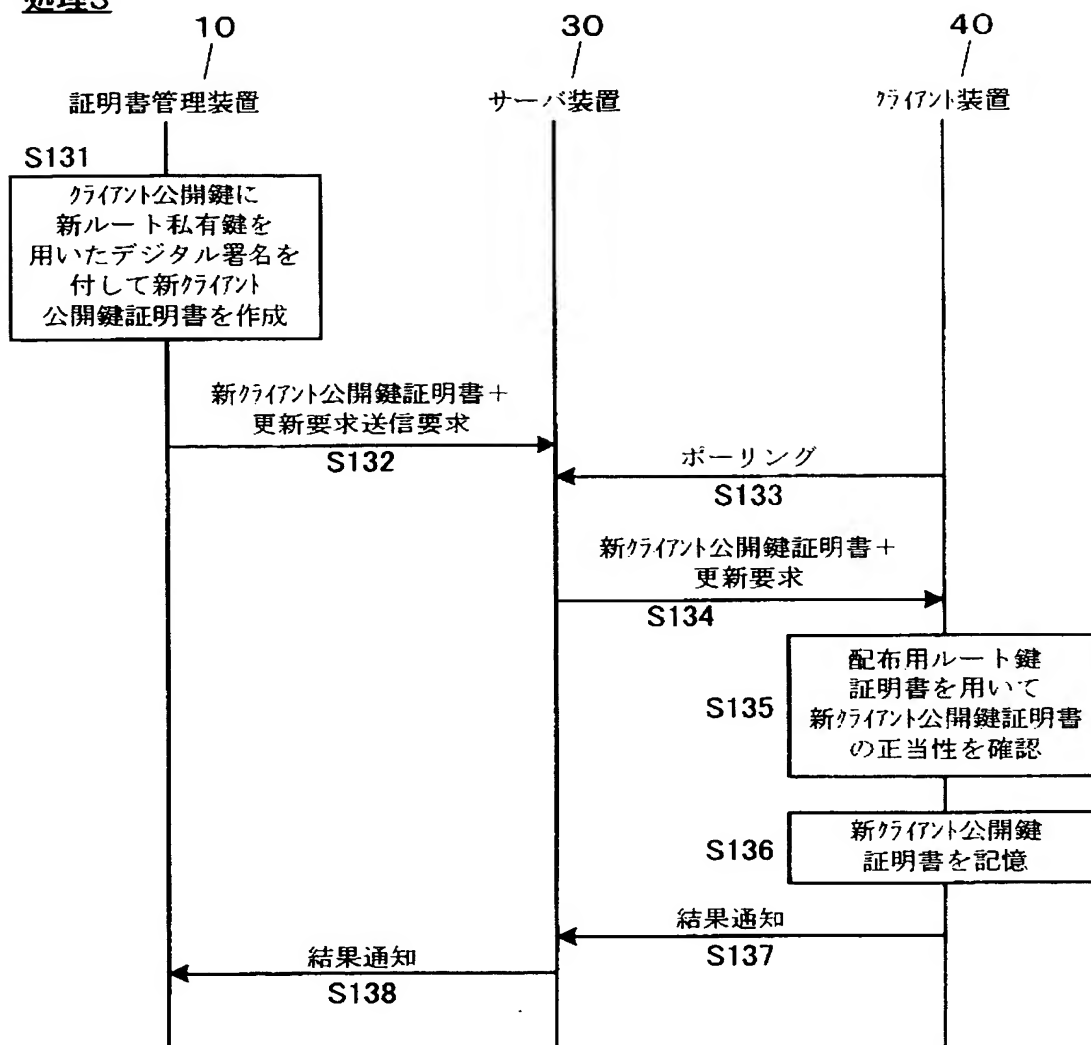
【図 6】

処理2

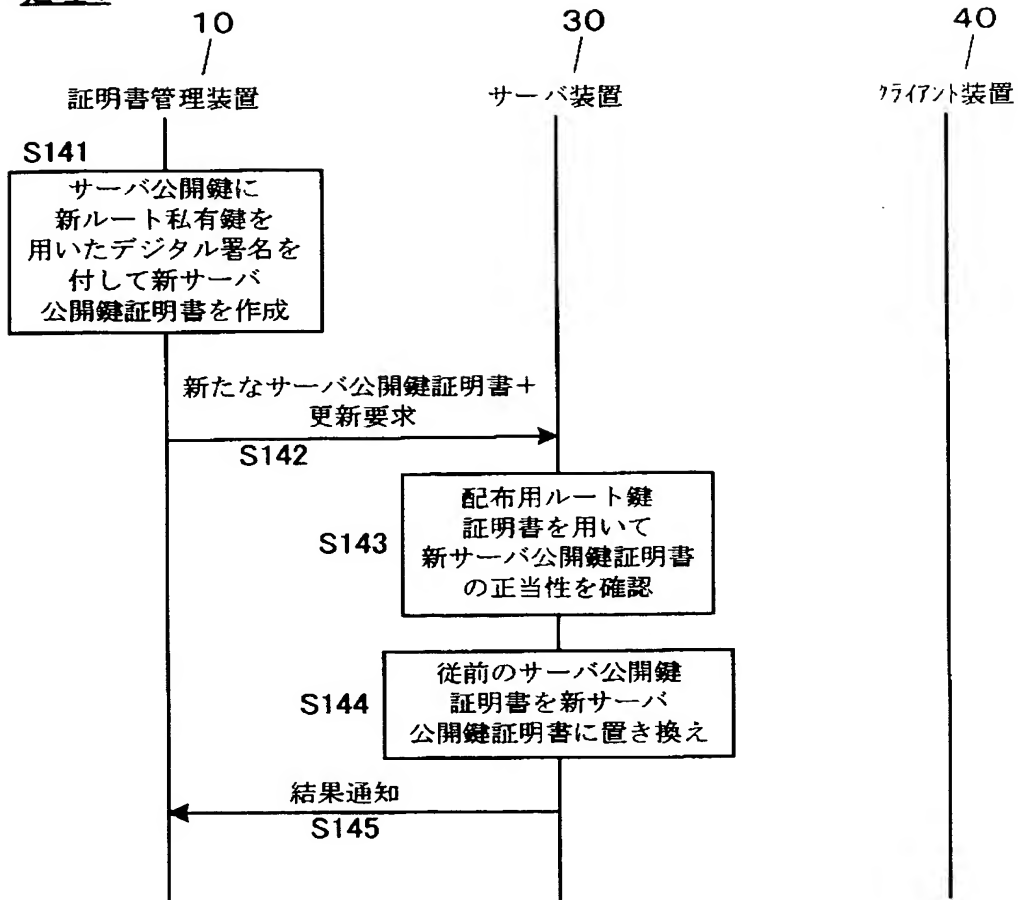


【図 7】

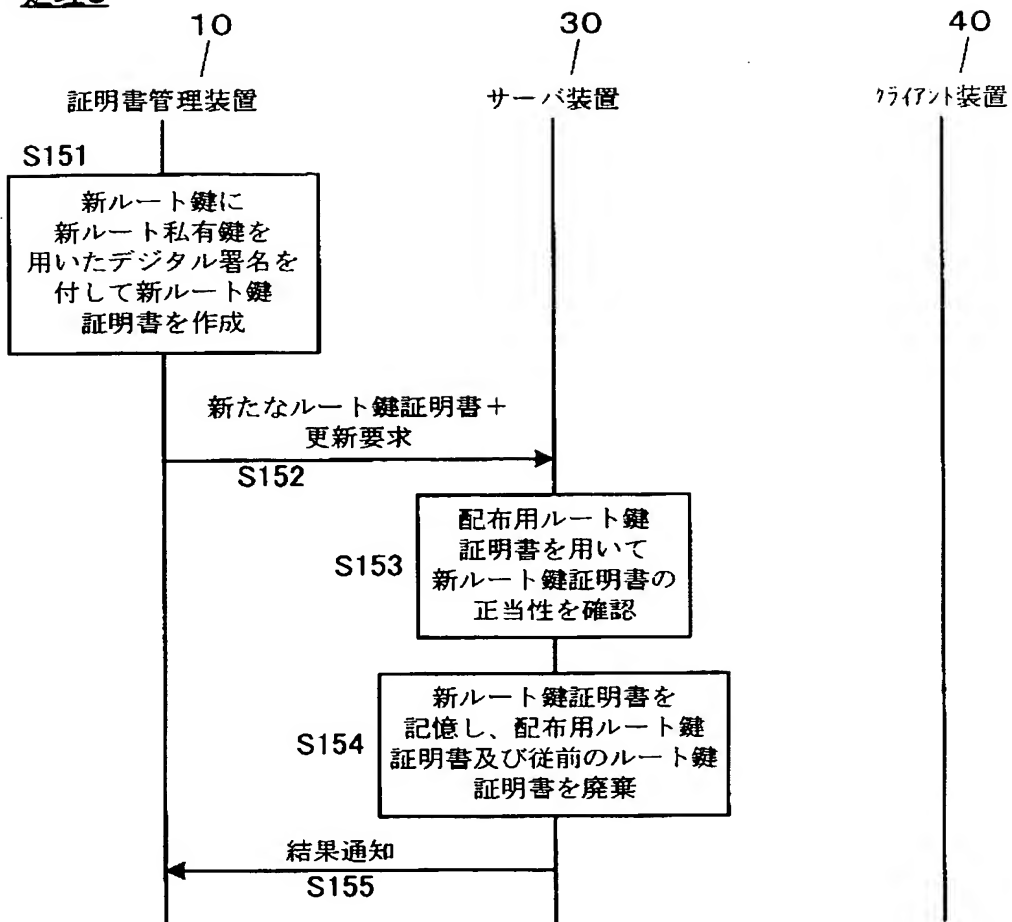
処理3



【図 8】

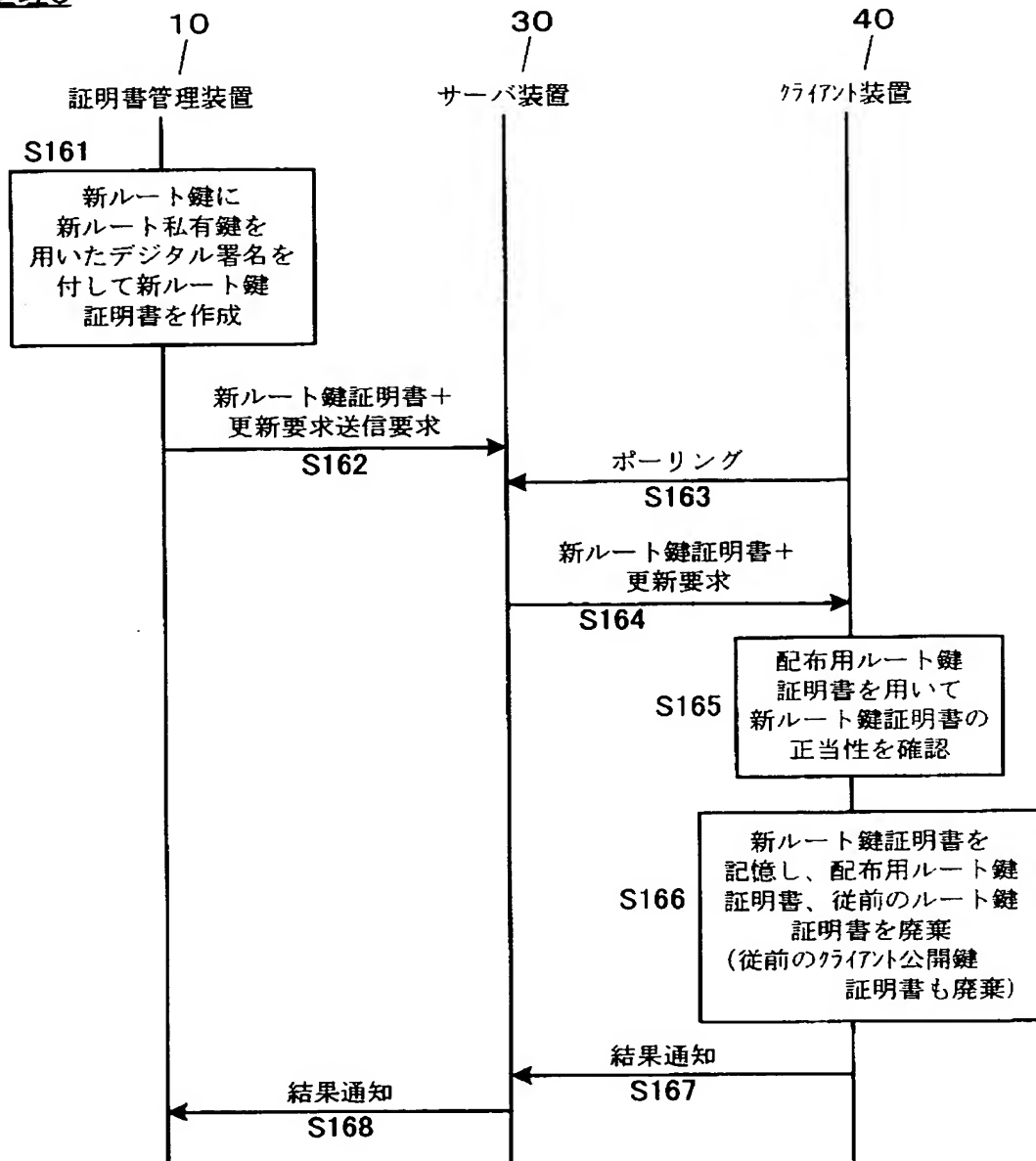
処理4

【図9】

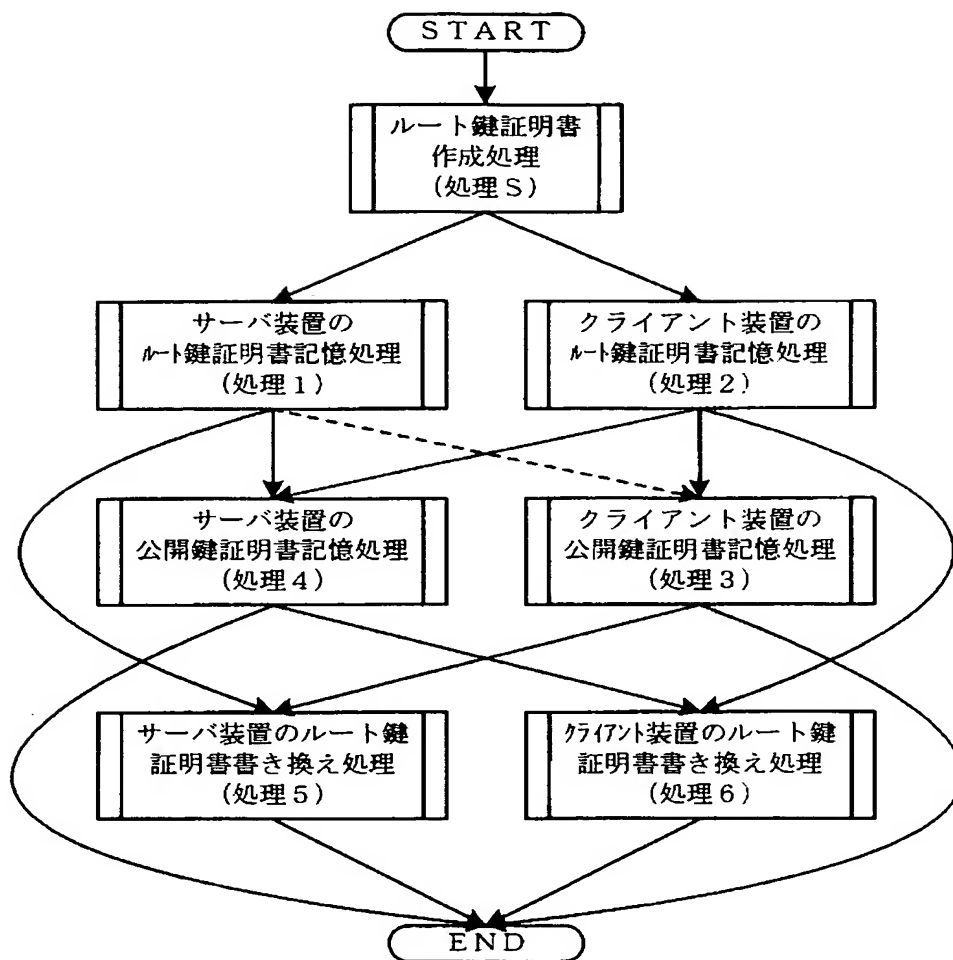
処理5

【図10】

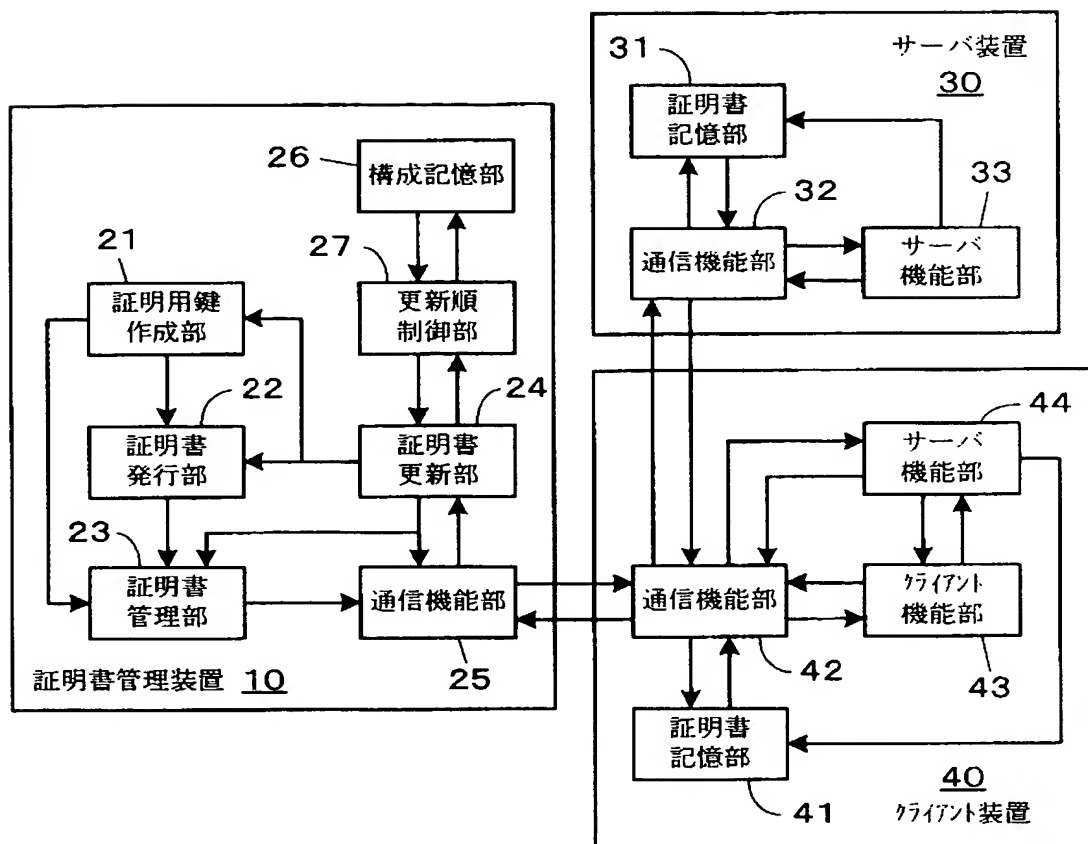
処理6



【図 11】

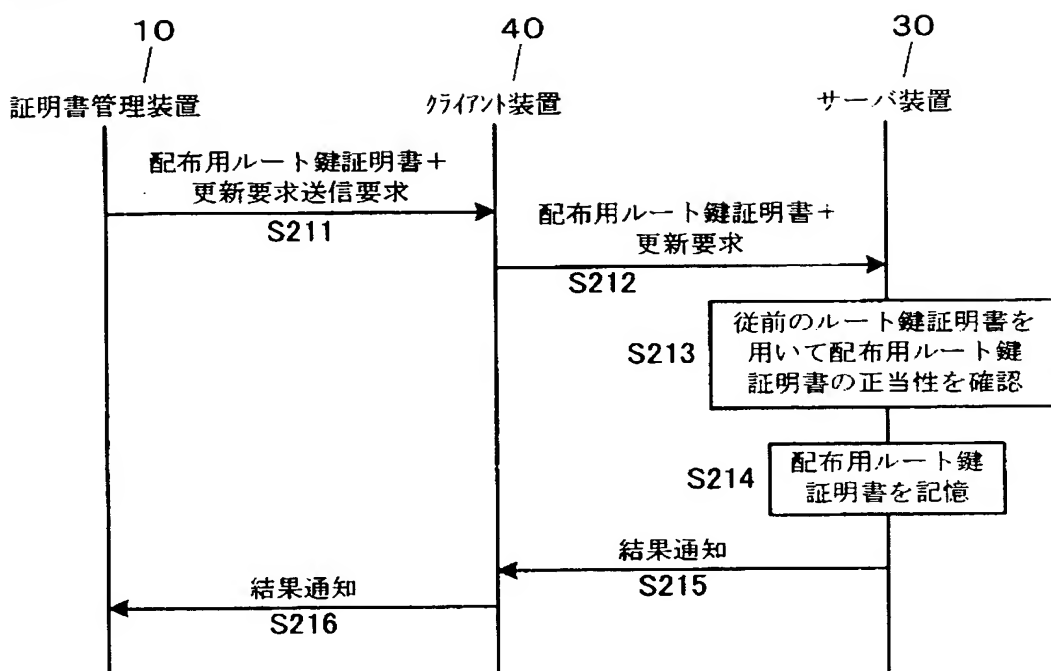


【図 12】



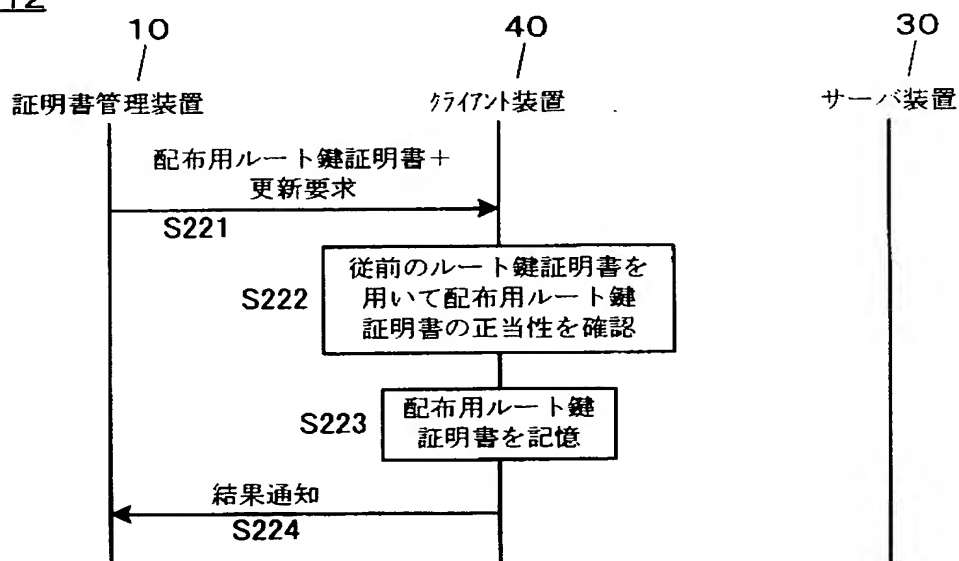
【図 13】

処理 11



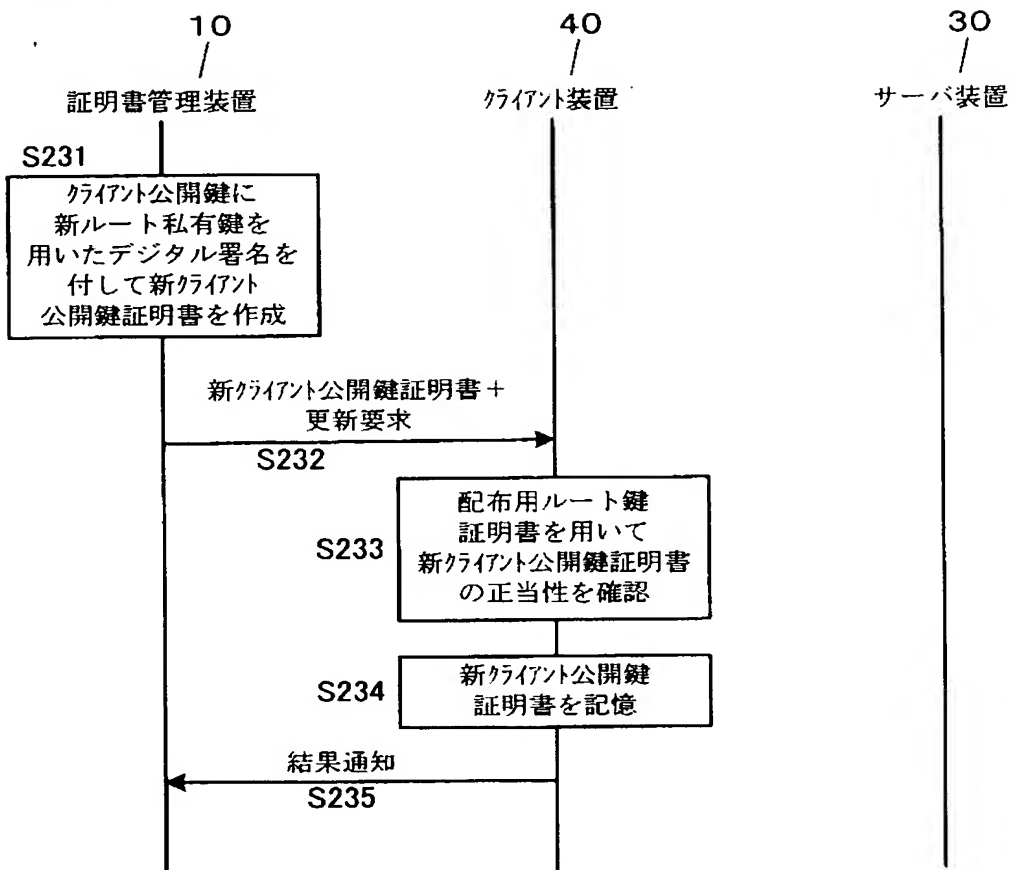
【図 14】

処理 12



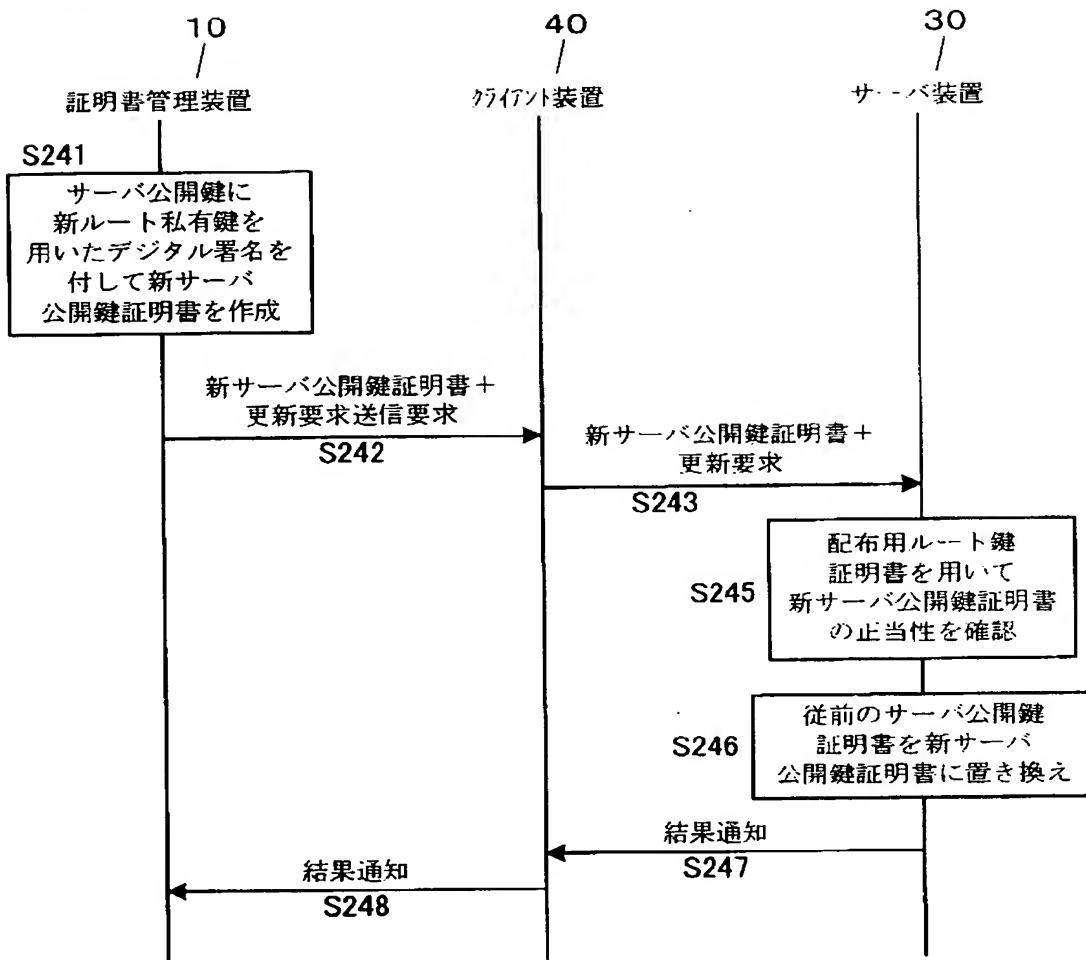
【図 15】

処理13



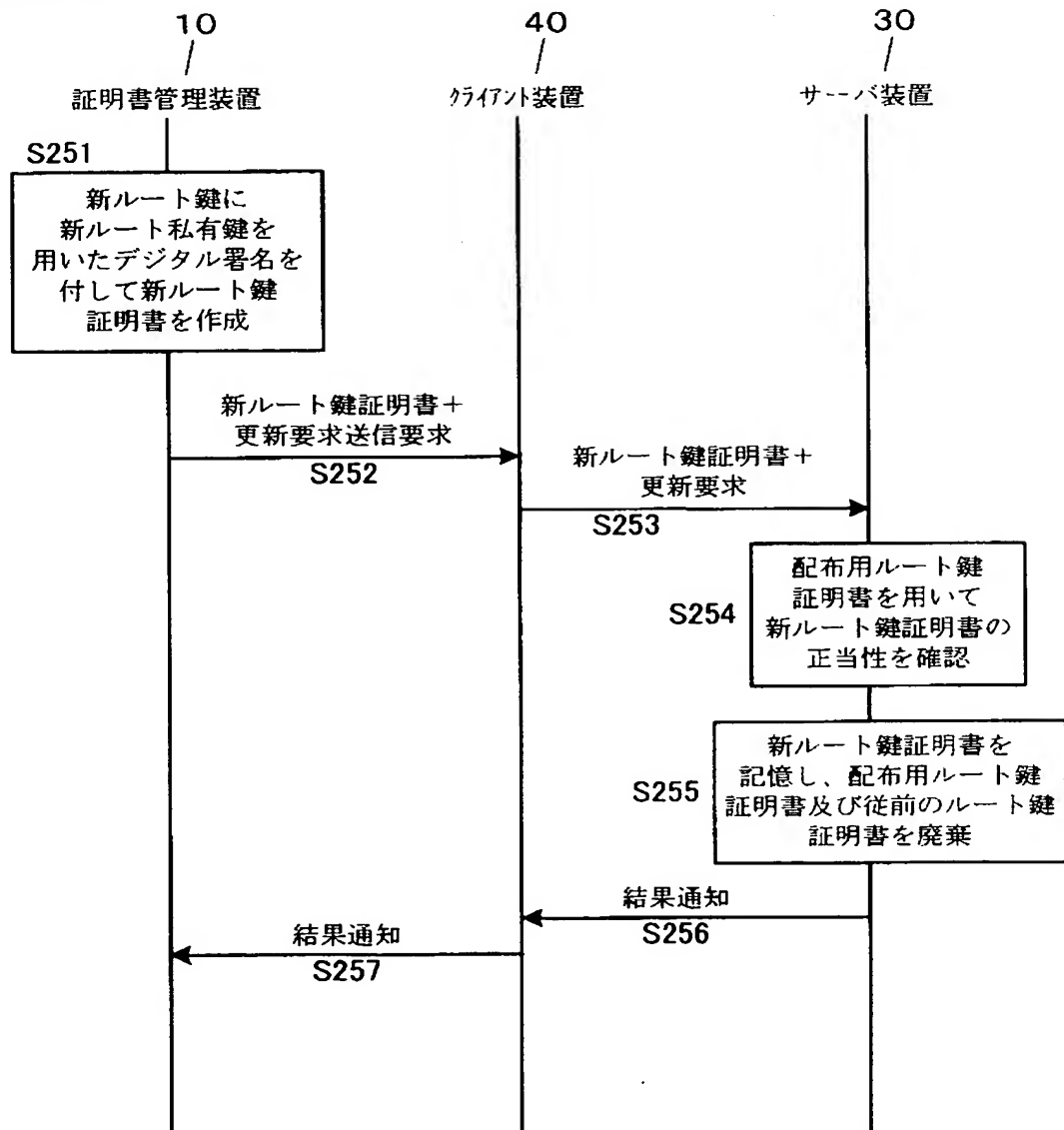
【図 16】

処理 14



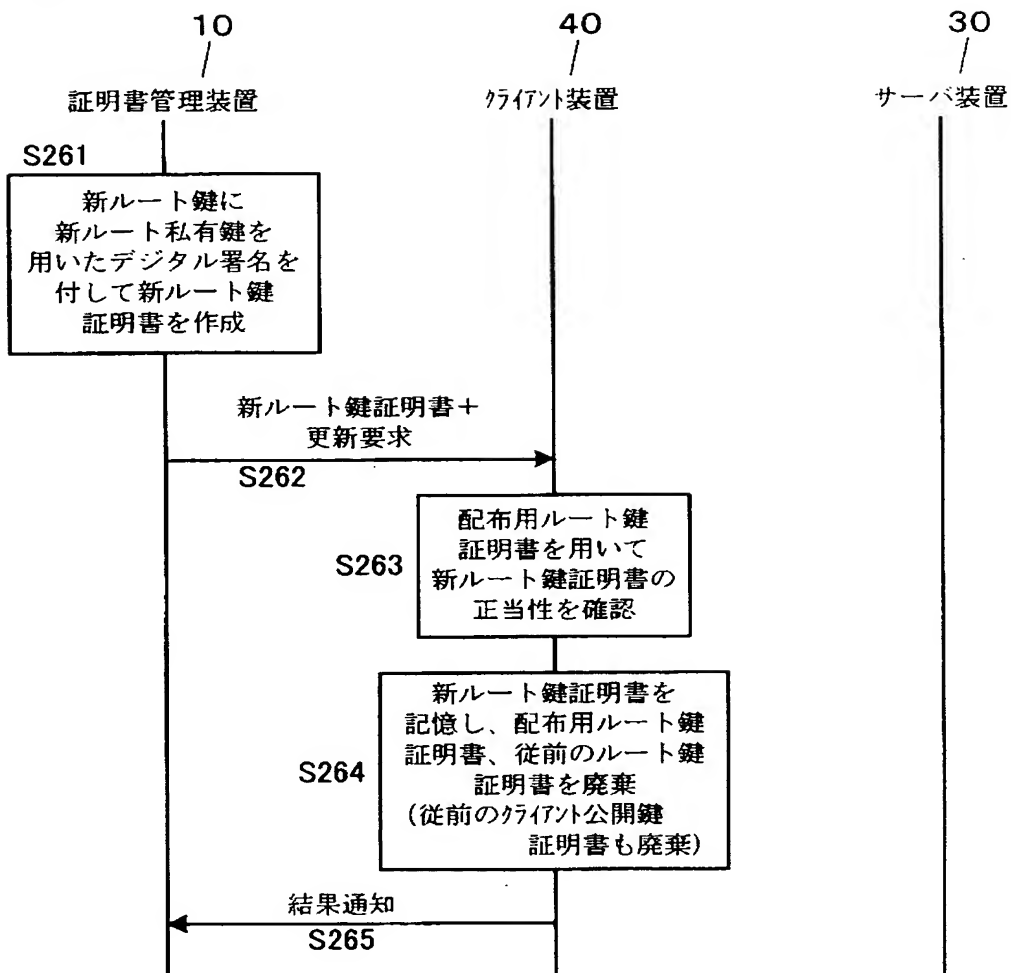
【図 17】

処理15

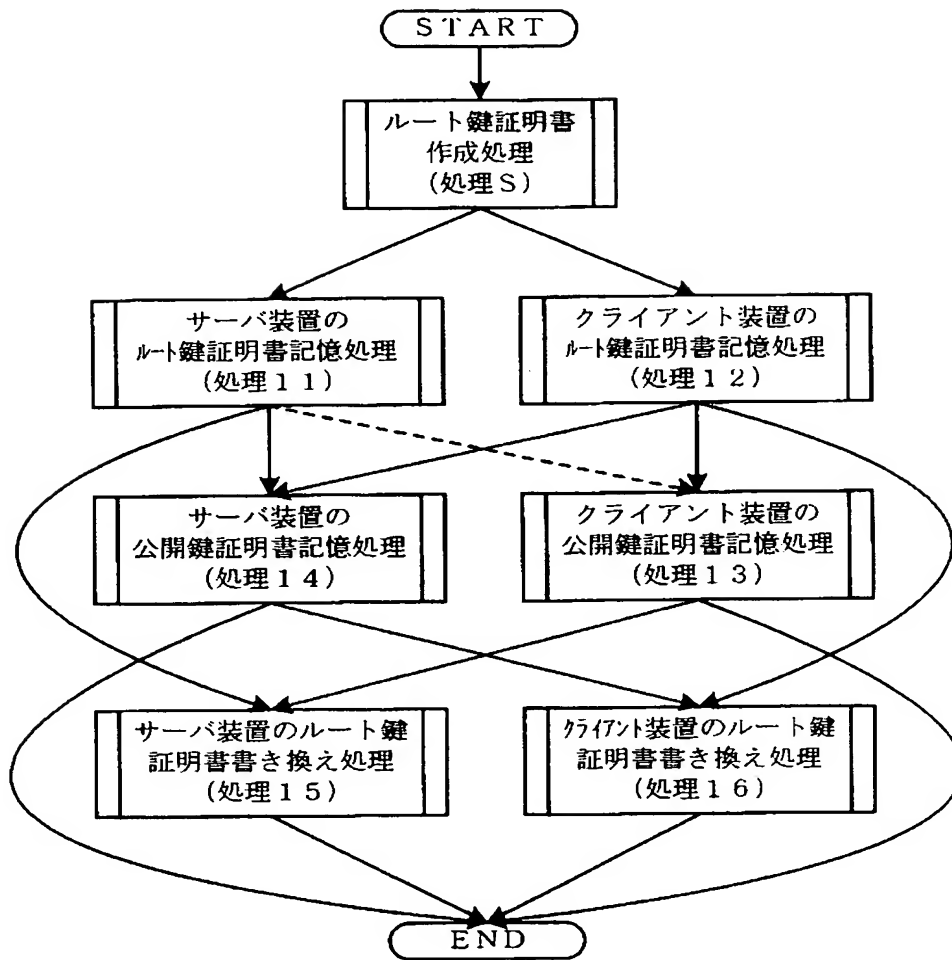


【図 18】

処理16

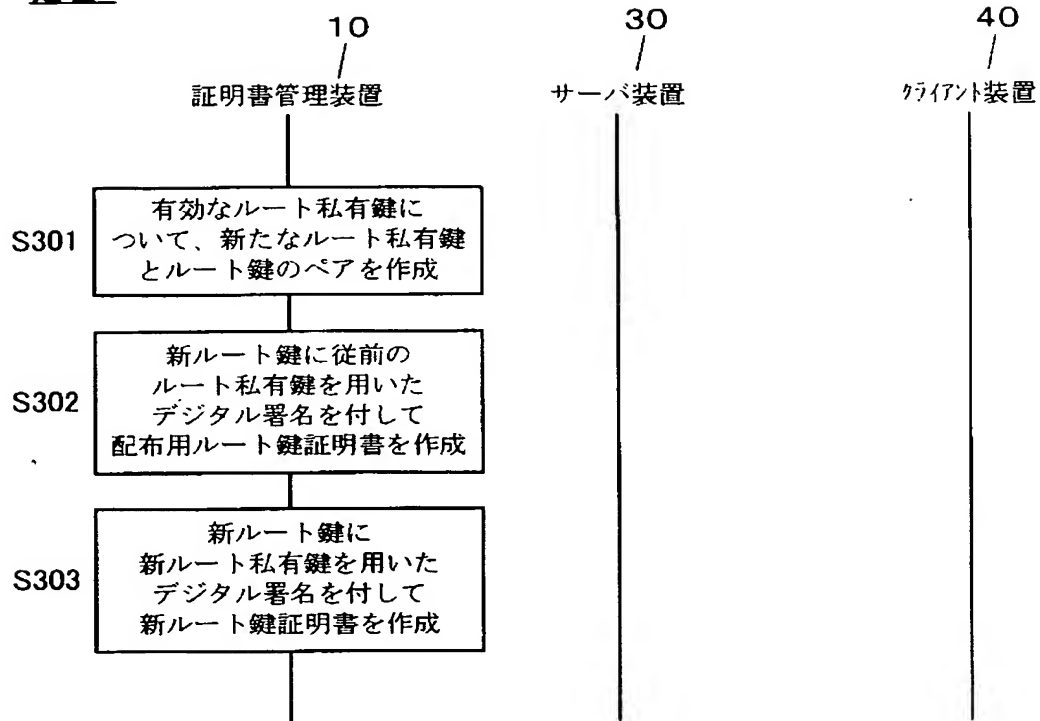


【図 19】



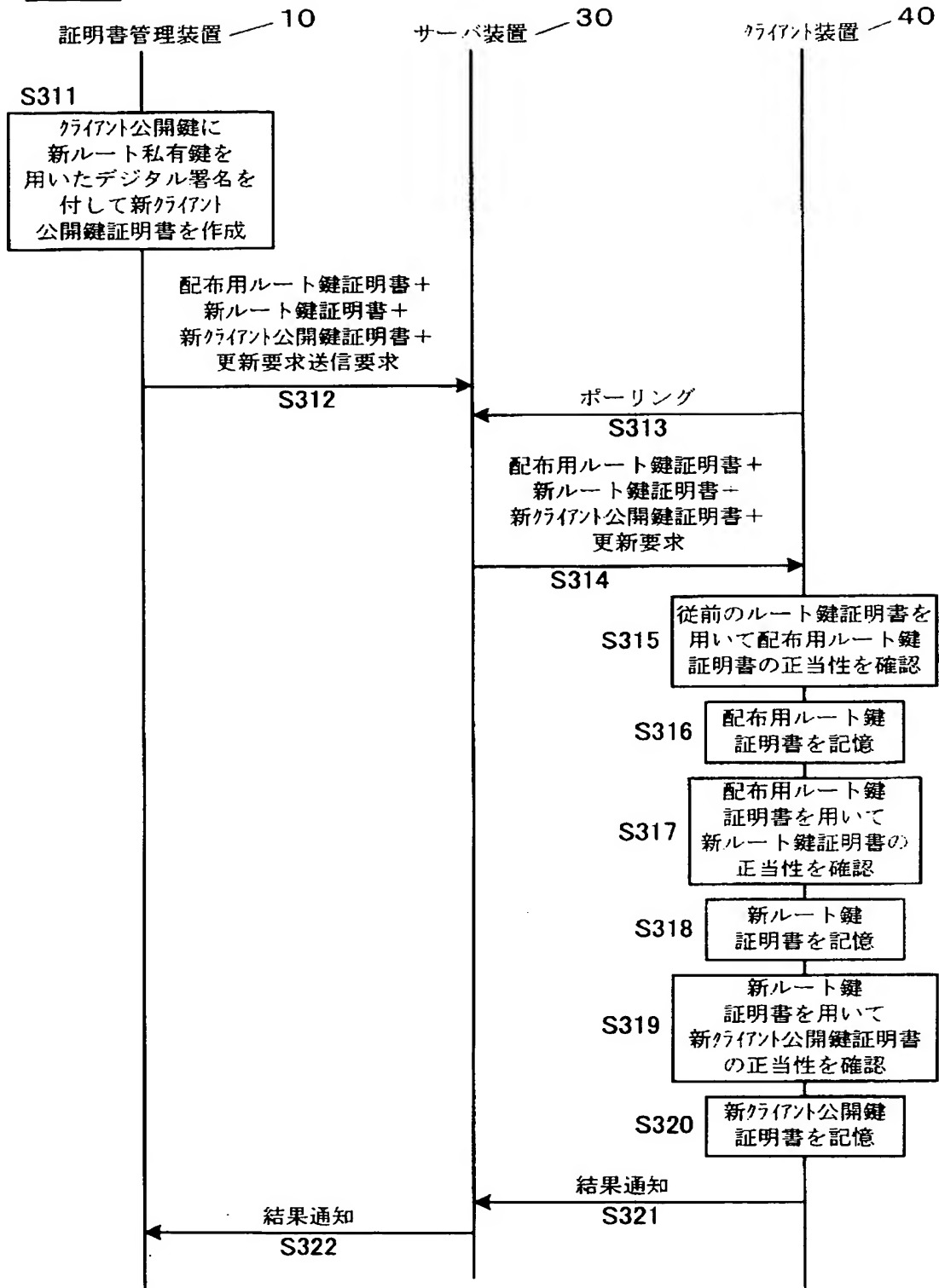
【図 20】

処理 I



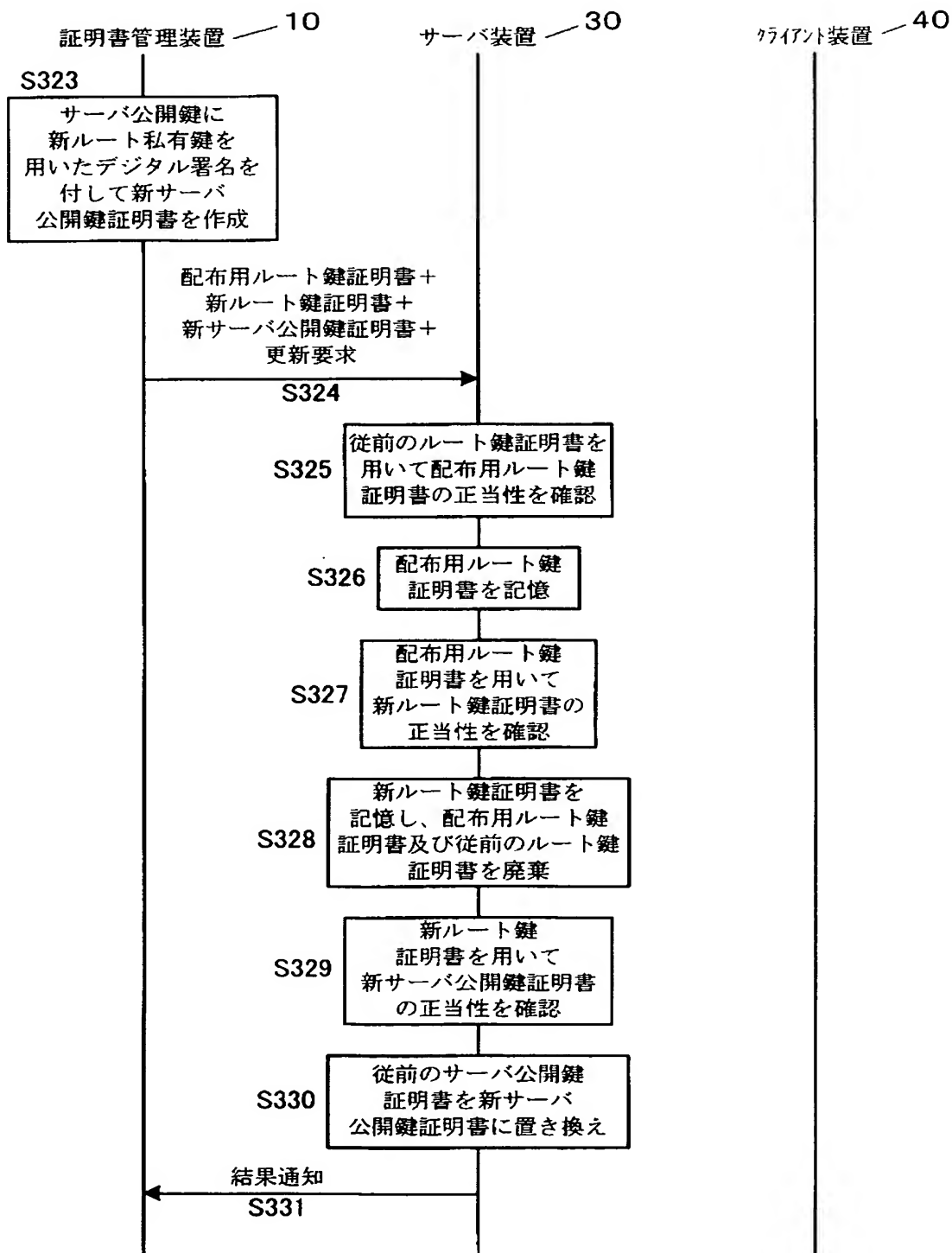
【図 21】

処理21

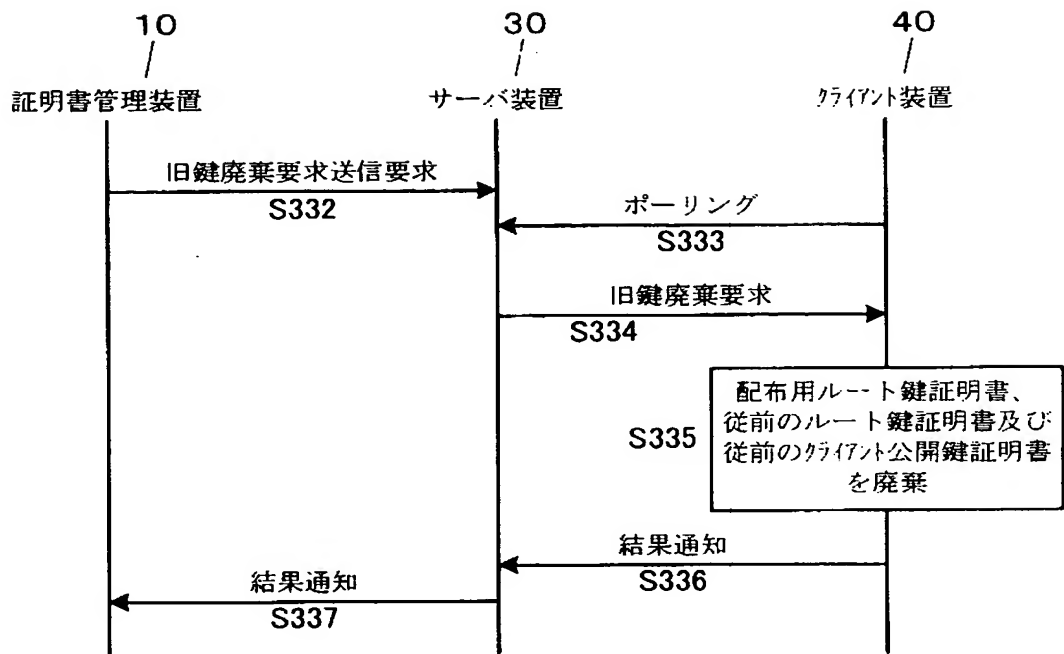


【図 22】

処理22

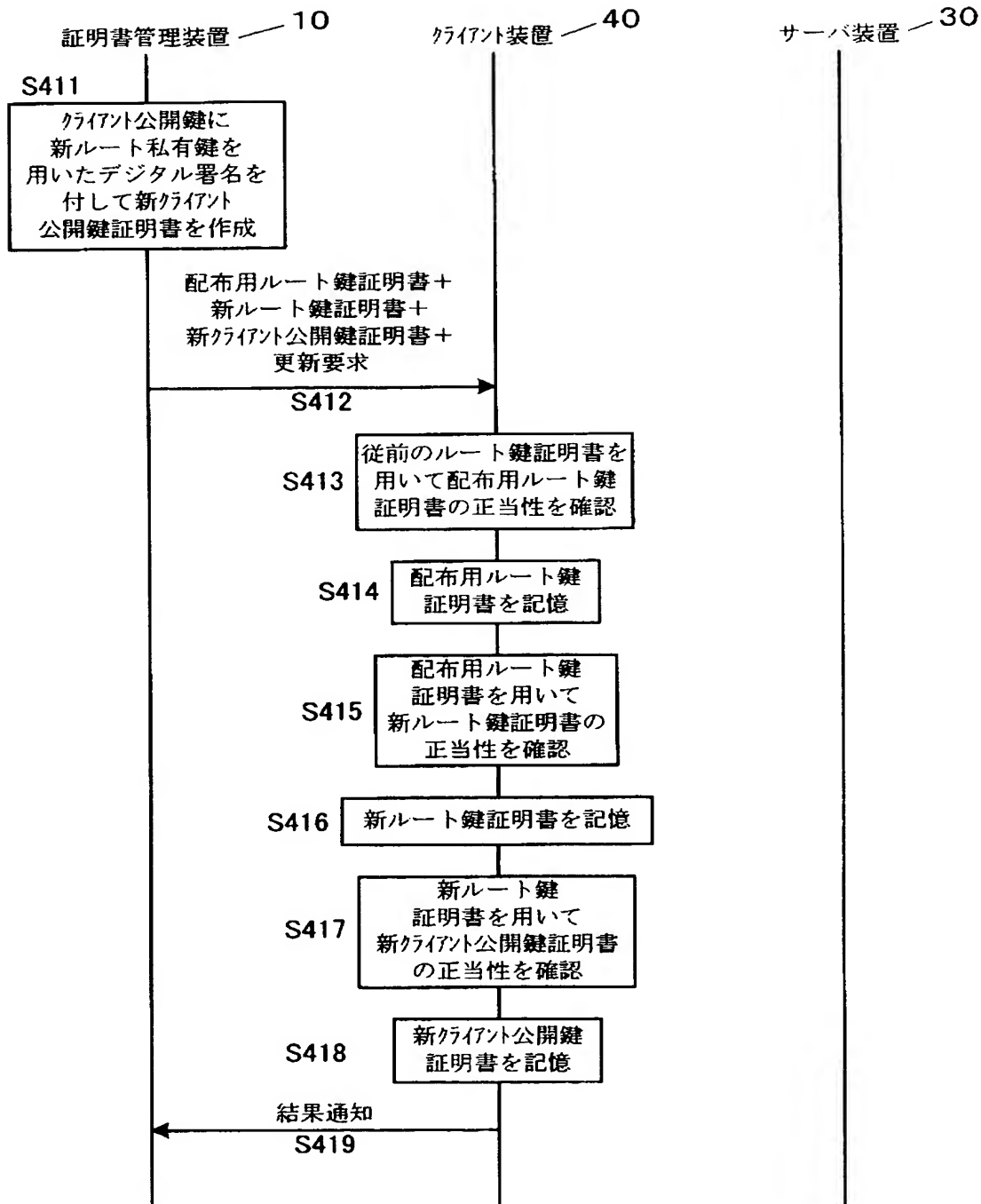


【図 23】

処理23

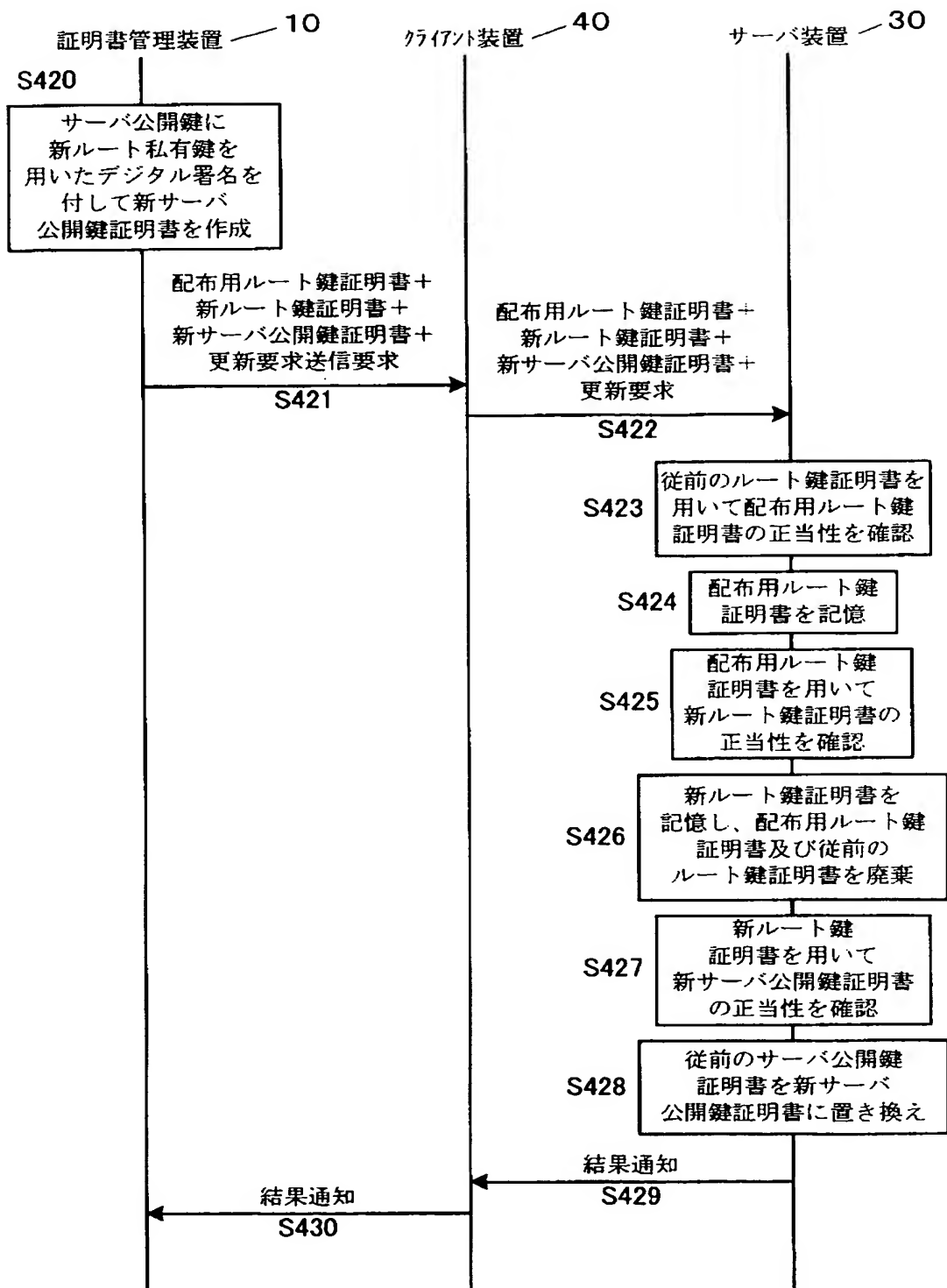
【図 24】

処理31

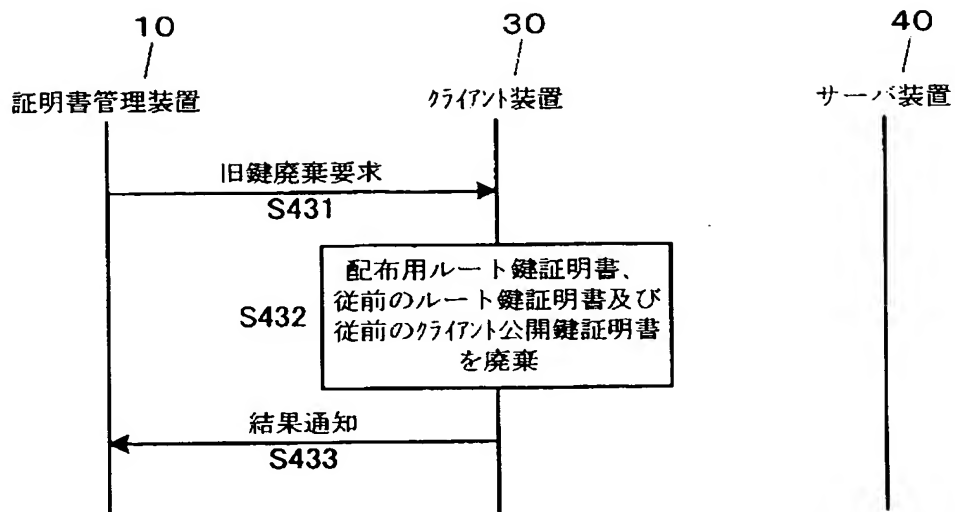


【図 25】

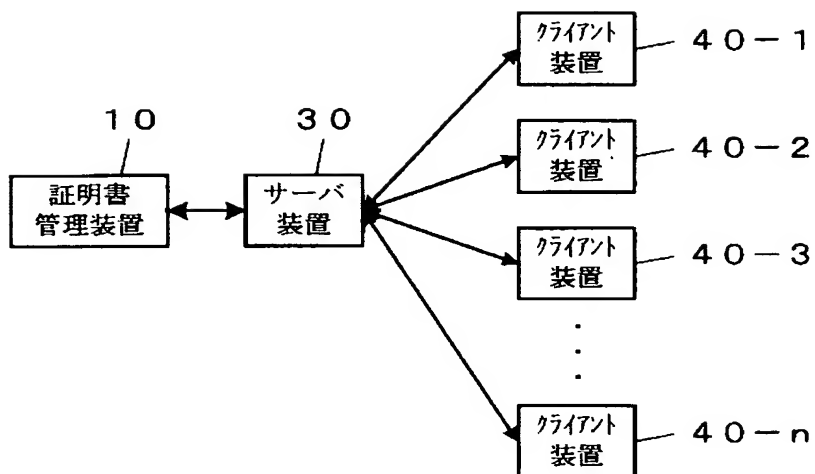
処理32



【図 26】

処理33

【図 27】



【図 28】

ノードID	
管理装置との直接通信可否	
通信相手 ノードID	クライアント/サーバ
	使用ルート鍵
	更新状態
	...
.	...
	...
	...
	...
.	...
	...
	...
	...

【図 29】

(a)

サーバ装置 30	
直接通信可	
クライアント 装置 40-1	サーバ
	ルート鍵 A
	更新要
クライアント 装置 40-2	サーバ
	ルート鍵 A
	更新要
クライアント 装置 40-3	サーバ
	ルート鍵 A
	更新要
.	...
	...
	...
クライアント 装置 40-n	サーバ
	ルート鍵 A
	更新要

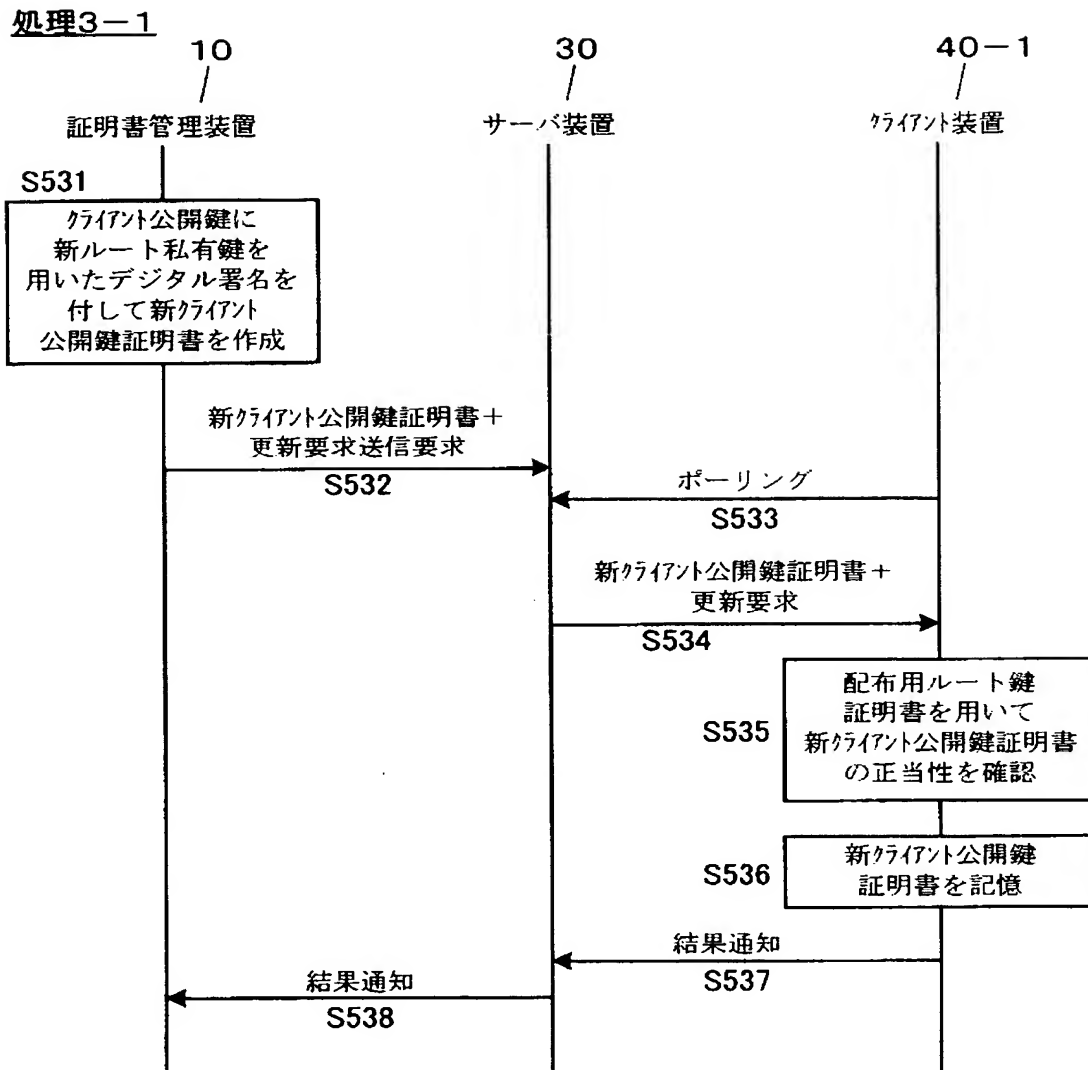
(b)

クライアント装置 40-1	
直接通信否	
なし	

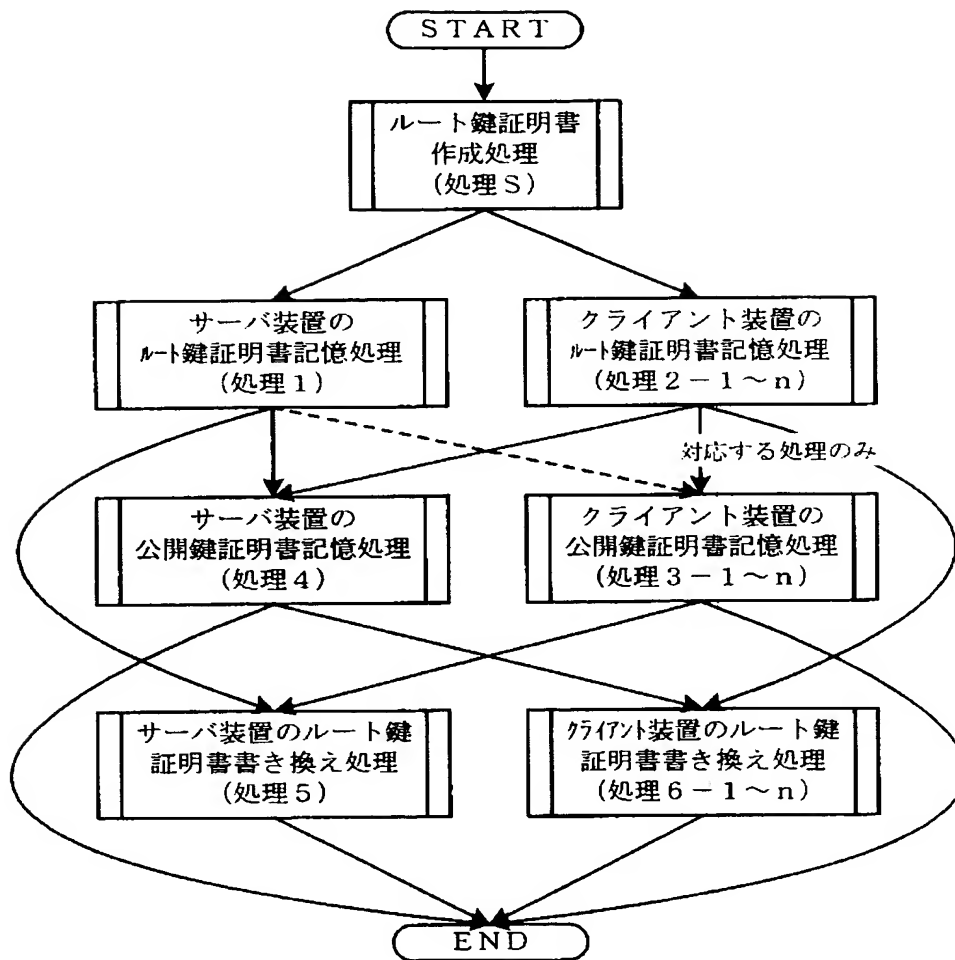
(c)

クライアント装置 40-1	
直接通信否	
サーバ 装置 30	クライアント
	ルート鍵 A
	更新要

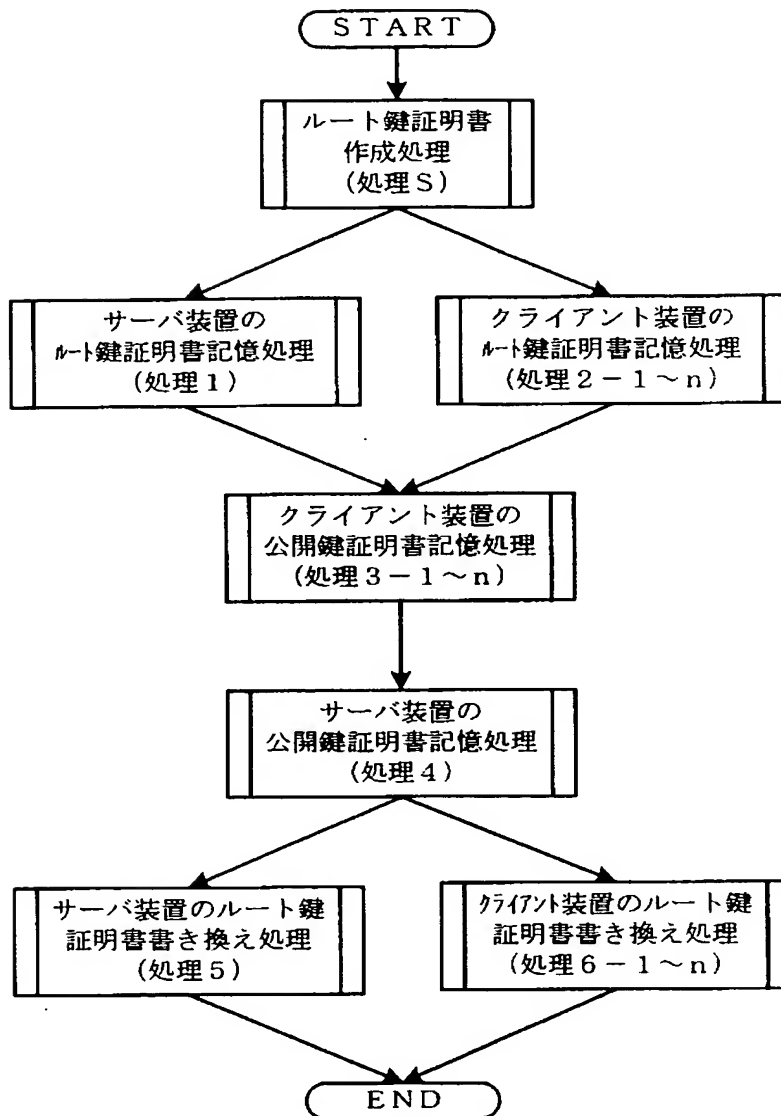
【図 30】



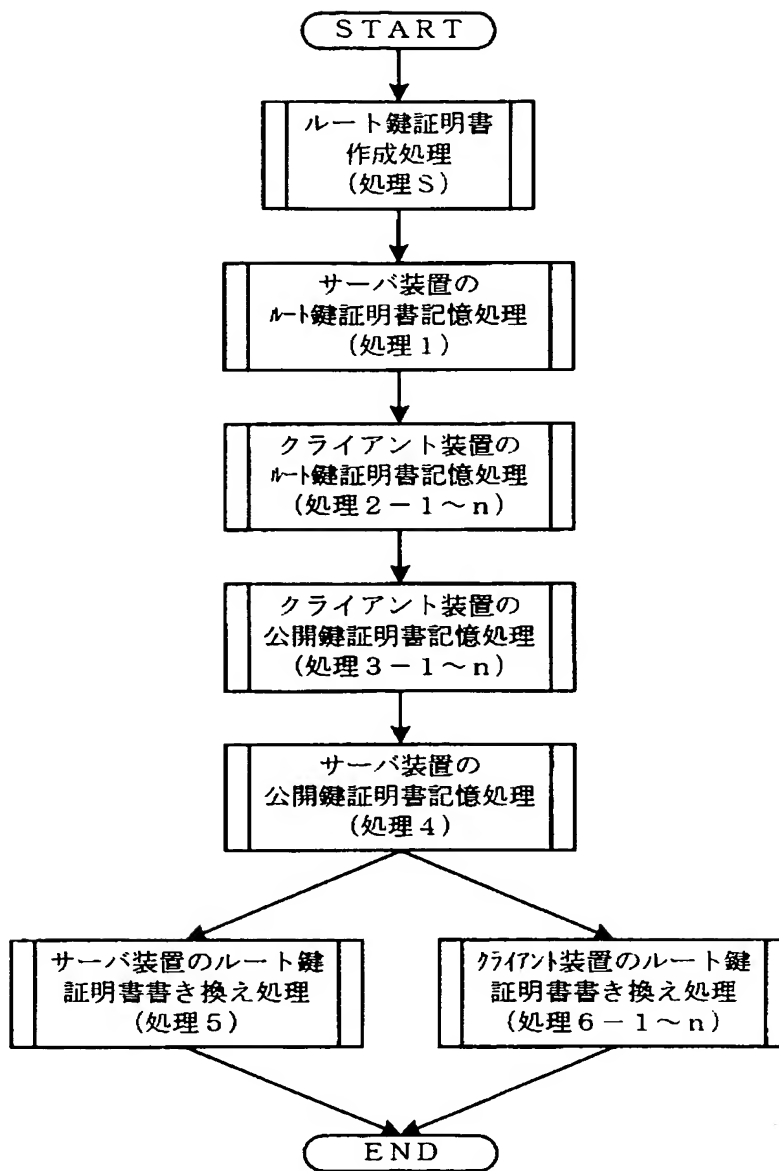
【図 31】



【図 3 2】

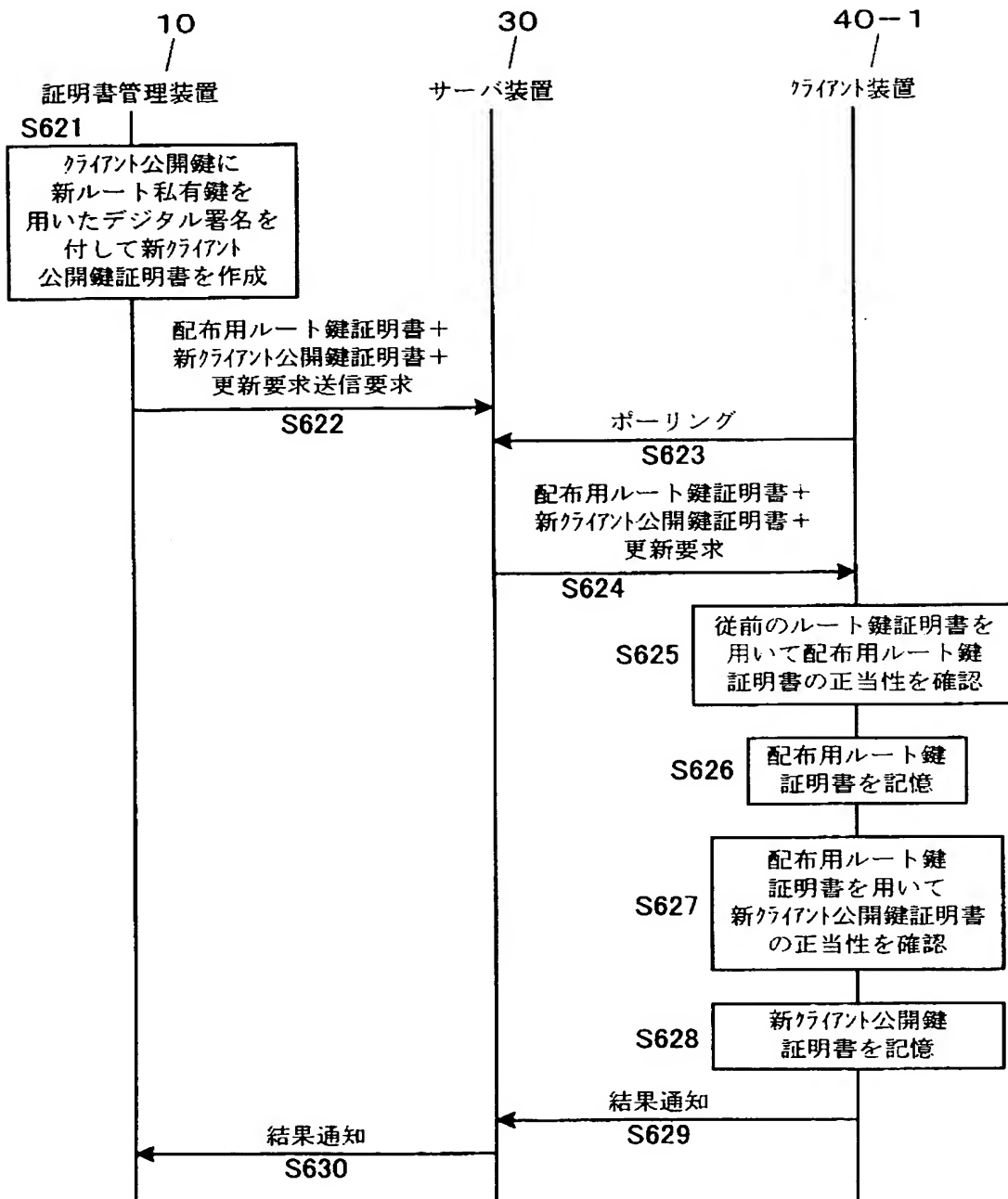


【図 33】

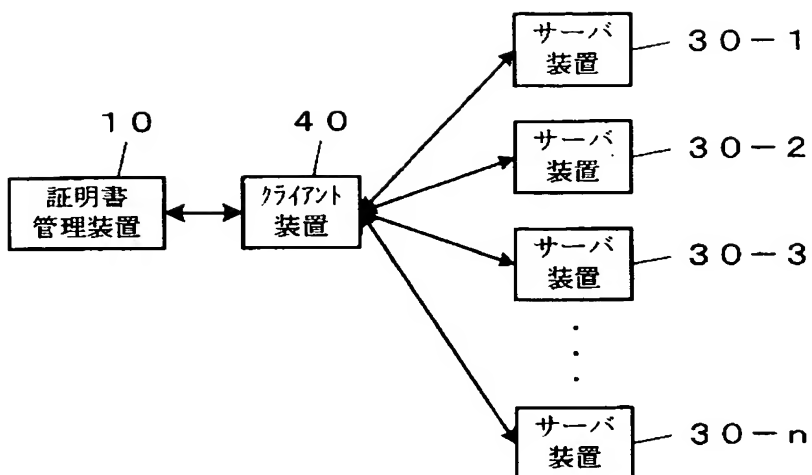


【図 34】

処理2' -1



【図 35】



【図 36】

(a)

クライアント装置 40	
直接通信可	
サーバ装置 30-1	クライアント
	ルート鍵A
	更新要
サーバ装置 30-2	クライアント
	ルート鍵A
	更新要
サーバ装置 30-3	クライアント
	ルート鍵A
	更新要
...	...
	...
	...
サーバ装置 30-n	クライアント
	ルート鍵A
	更新要

(b)

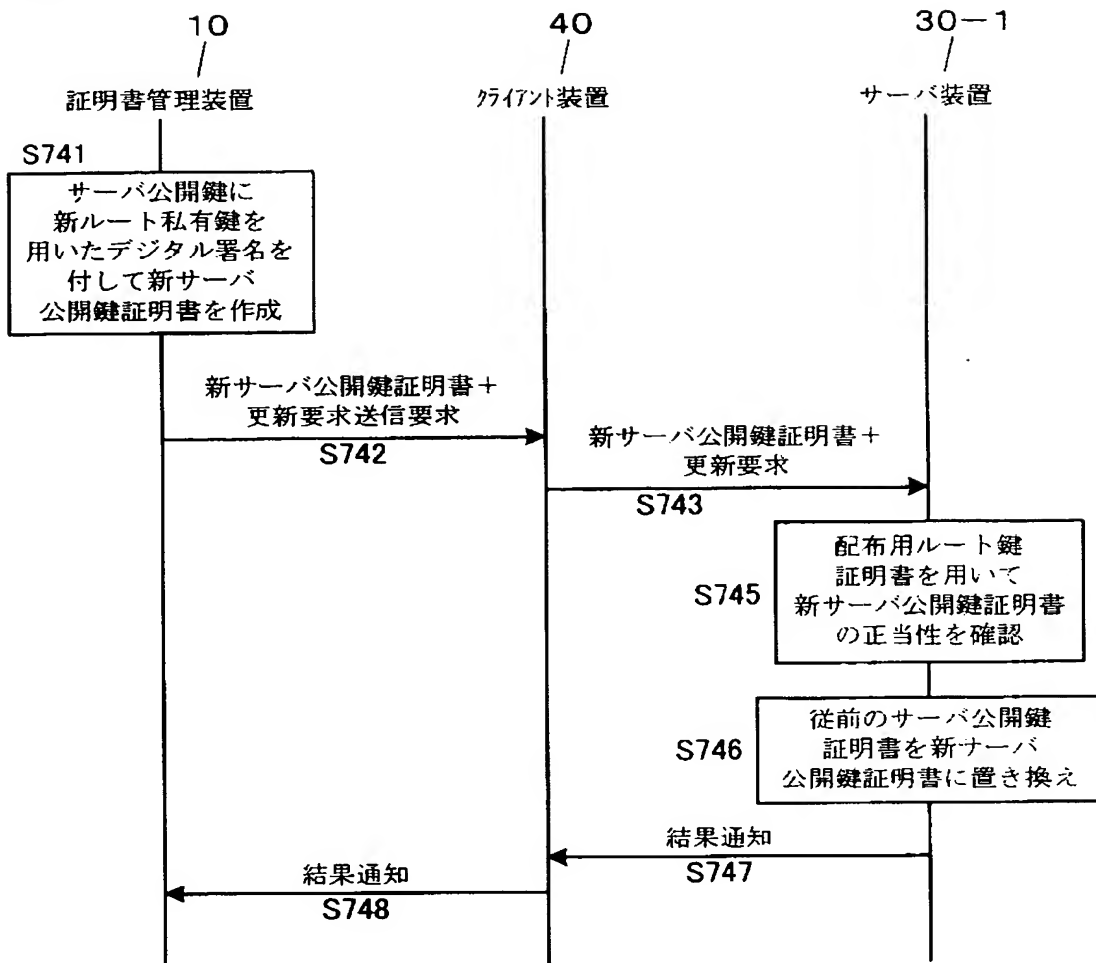
サーバ装置 30-1	
直接通信否	
なし	

(c)

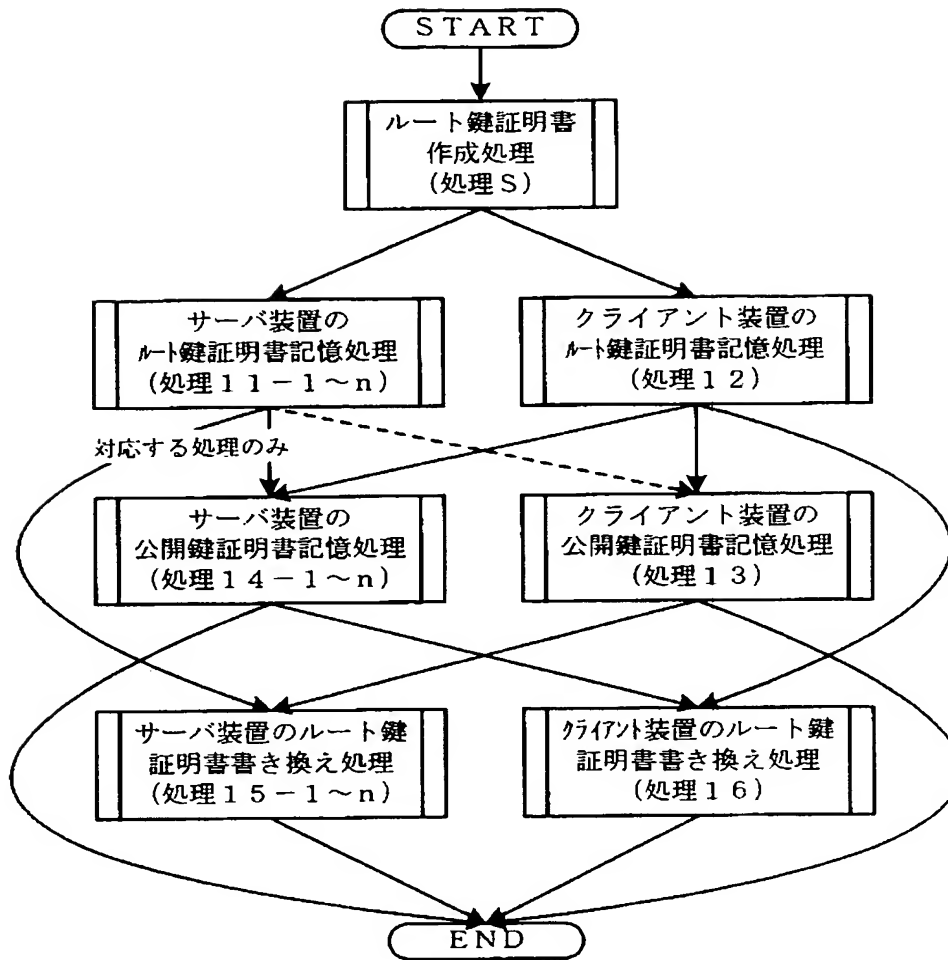
サーバ装置 30-1	
直接通信否	
クライアント装置 30	サーバ
	ルート鍵A
	更新要

【図 37】

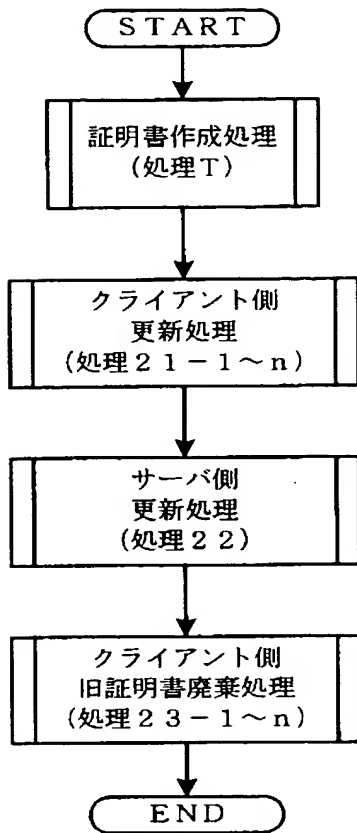
処理 14-1



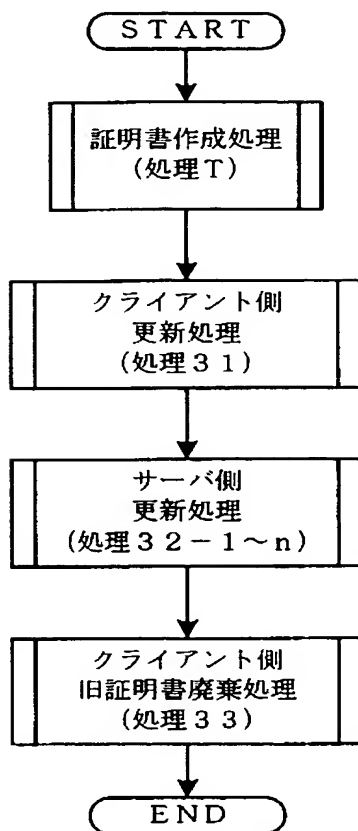
【図 38】



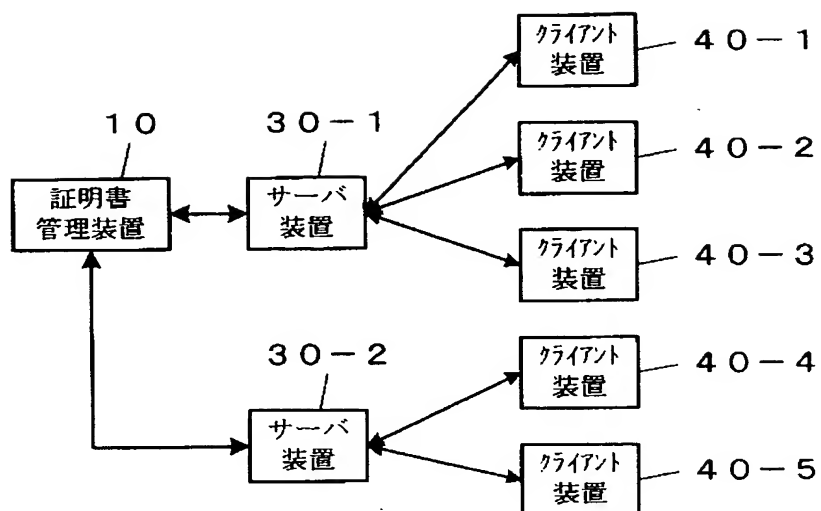
【図 39】



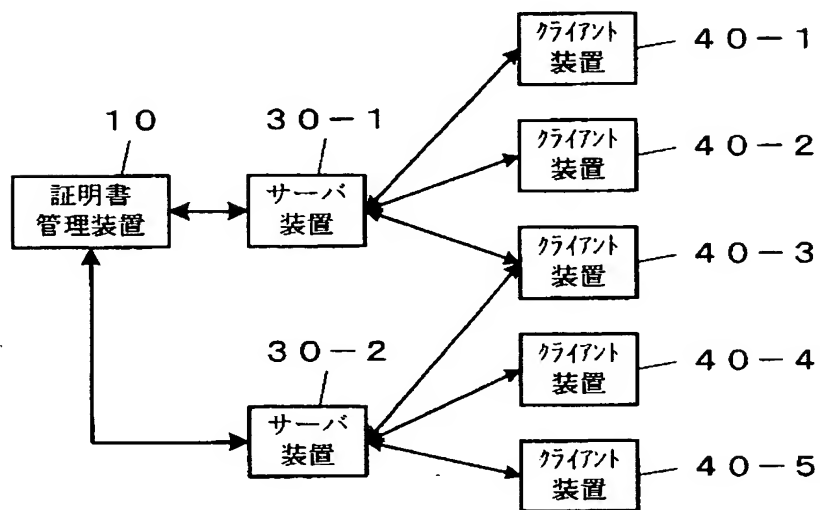
【図 4 0】



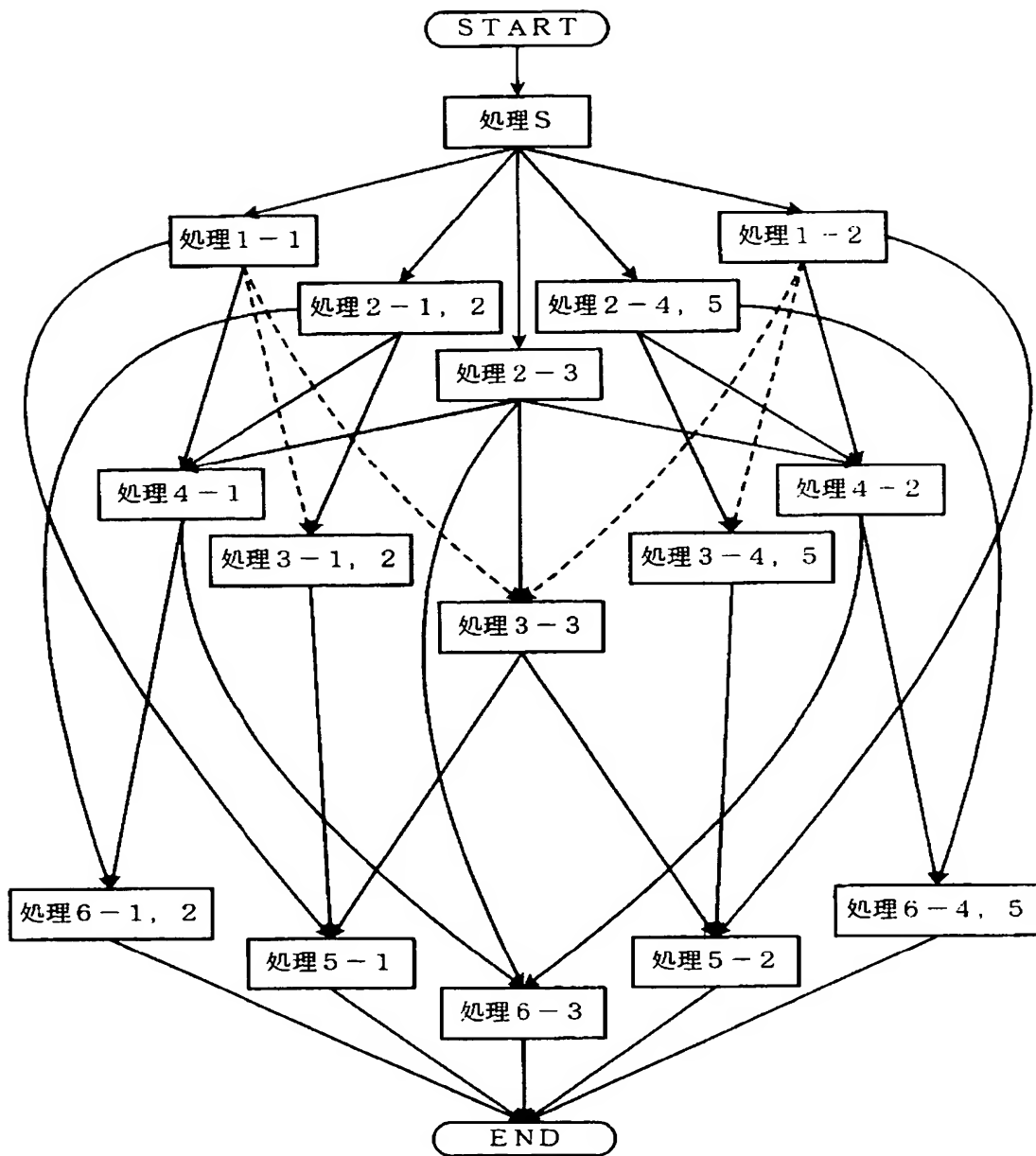
【図 4 1】



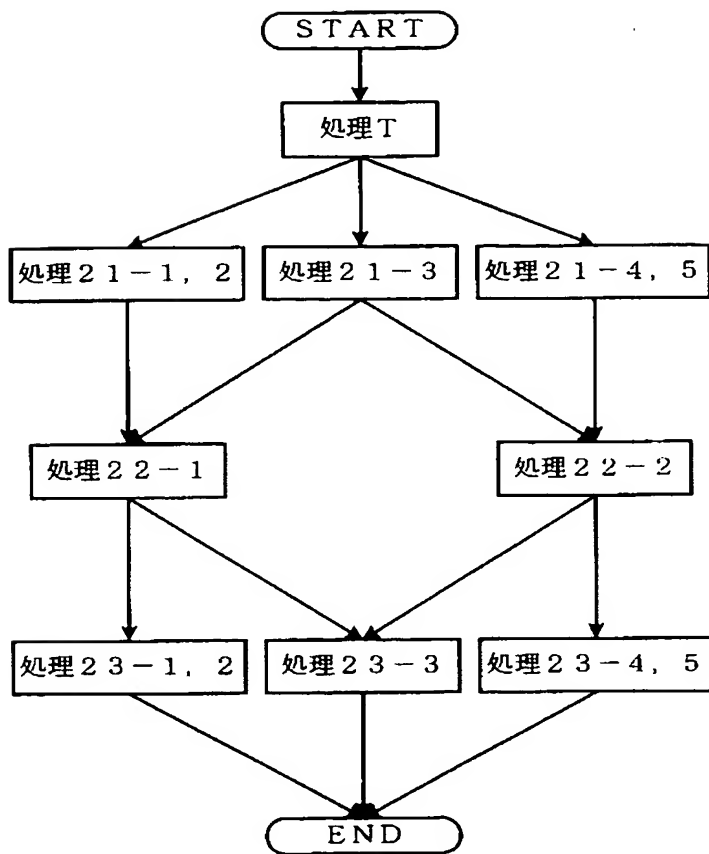
【図 4 2】



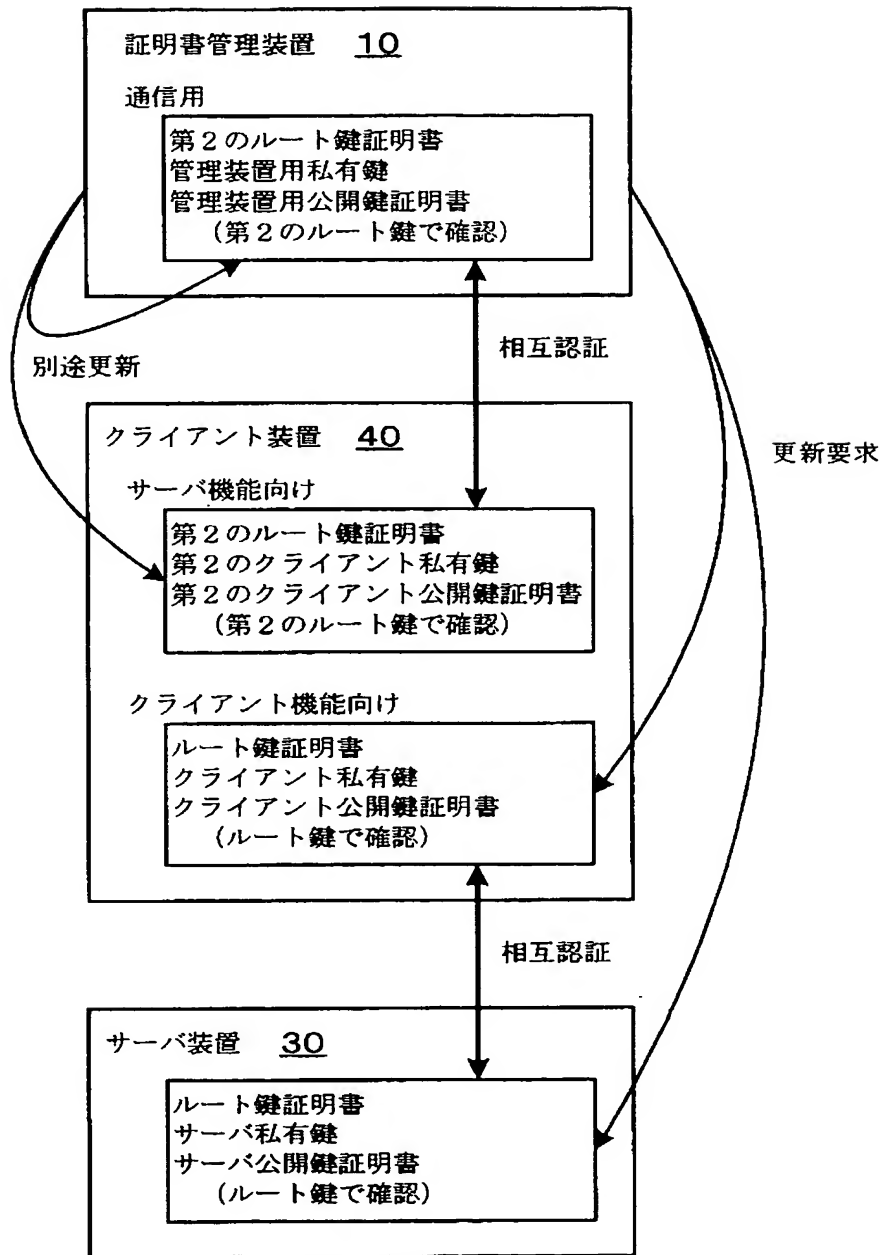
【図 43】



【図 44】



【図 45】

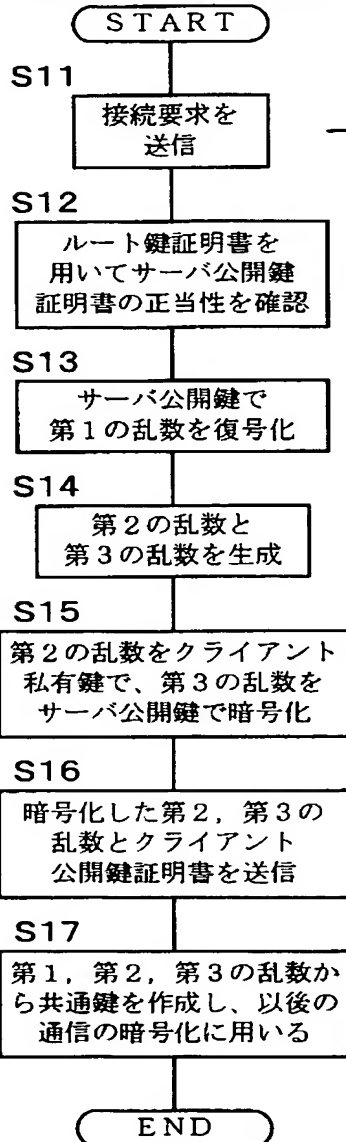


【図 4 6】

クライアント装置側

ルート鍵証明書
クライアント私有鍵
クライアント公開鍵証明書

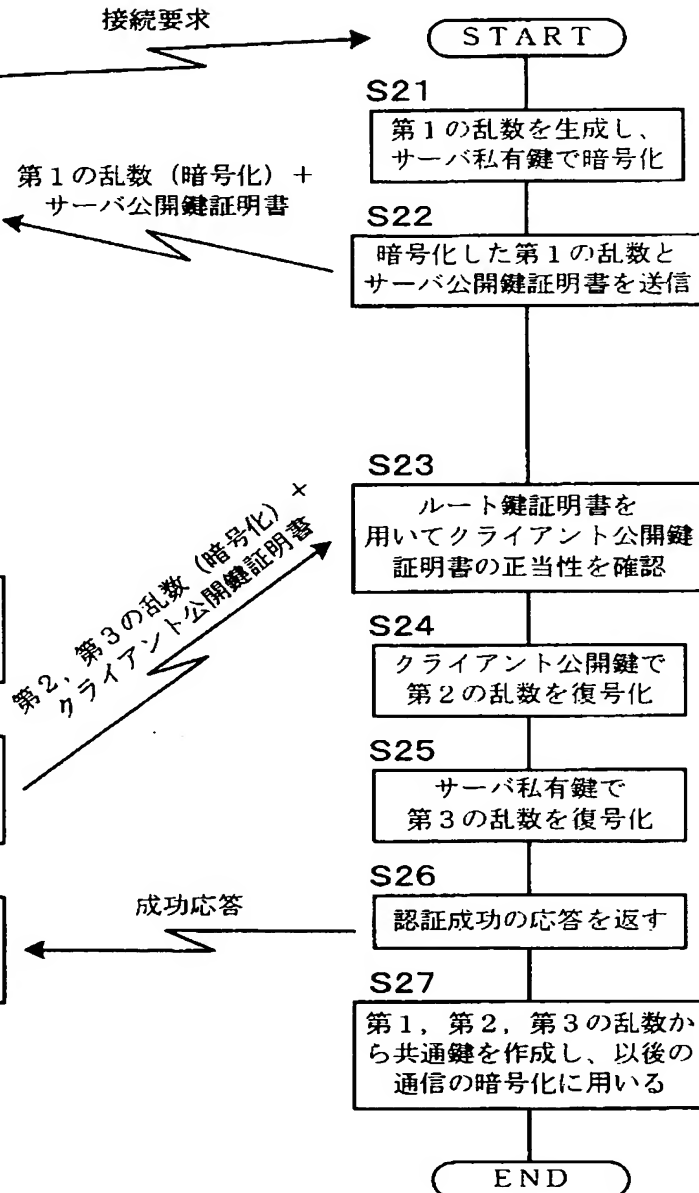
クライアント装置側処理



サーバ装置側

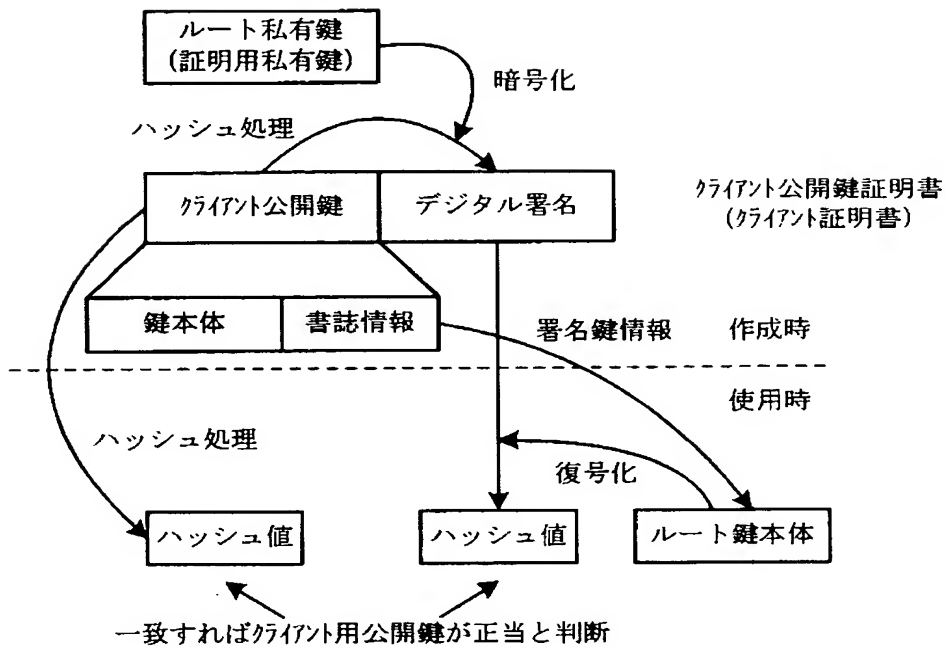
ルート鍵証明書
サーバ私有鍵
サーバ公開鍵証明書

サーバ装置側処理

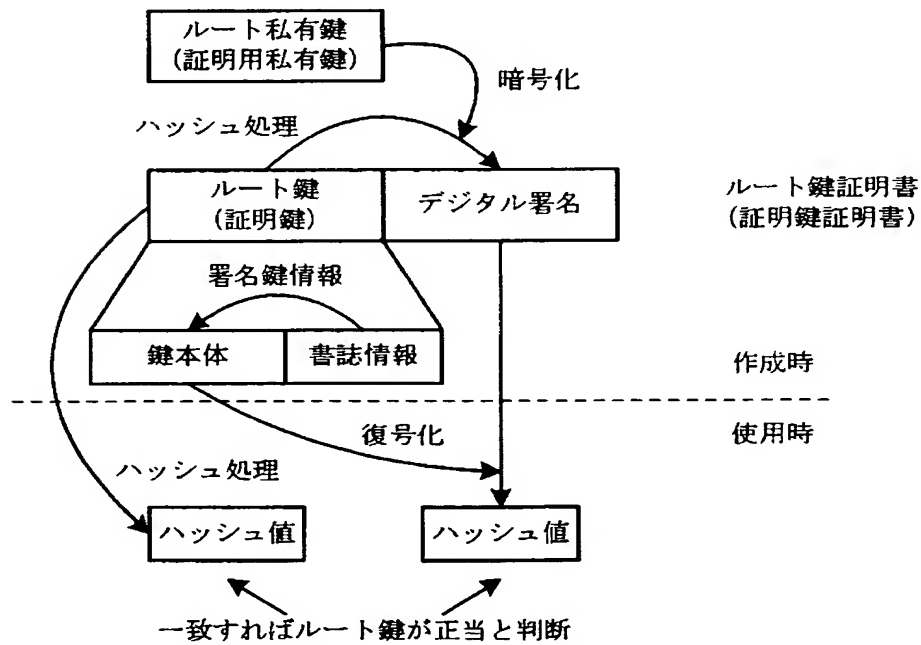


【図 47】

(a)



(b)



【書類名】 要約書

【要約】

【課題】 クライアント・サーバシステムにおける認証処理でデジタル証明書の正当性の確認に用いるルート鍵を自動的に更新できるようにする。

【解決手段】 クライアント装置とサーバ装置との間で公開鍵暗号を利用したデジタル証明書を用いる S S L 等の方式による相互認証を行うようにしたクライアント・サーバシステムに、デジタル証明書管理装置を接続し、サーバ装置とクライアント装置のルート鍵を自動的に更新するデジタル証明書管理システムを構成する。そして、この更新処理において、サーバ装置の公開鍵証明書を更新する処理（処理 4）を、そのサーバ装置の通信相手となる全てのクライアント装置について新ルート鍵を記憶させる処理（処理 2 - 1 ~ n）が完了した後で行うようにする。

【選択図】 図 3 1

特願 2 0 0 3 - 0 9 6 1 2 9

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 6 7 4 7]

1. 変更年月日 2 0 0 2 年 5 月 1 7 日

[変更理由] 住所変更

住 所 東京都大田区中馬込 1 丁目 3 番 6 号

氏 名 株式会社リコー